PB85-214450

Plume2D: Two-Dimensional Plumes in Uniform Ground Water Flow

Oklahoma State Univ., Stillwater

Prepared for

Robert S. Kerr Environmental Research Lab.
Ada, OK

Jun 85

U.S. Department of Commerce
National Technical Information Service
NTIS

PB85-214450

EPA/600/2-85/065
June 1985

PLUME2D

TWO-DIMENSIONAL PLUMES IN UNIFORM GROUND WATER FLOW

by

Jan Wagner and Stephanie A. Watts
School of Chemical Engineering
Oklahoma State University
Stillwater, Oklahoma 74078

Douglas C. Kent
Department of Geology
Oklahoma State University
Stillwater, Oklahoma 74078

CR811142

Project Officer

Carl G. Enfield
Robert S. Kerr Environmental Research Laboratory
U. S. Environmental Protection Agency
Ada, Oklahoma 74820

ROBERT S. KERR ENVIRONMENTAL RESEARCH LABORATORY
OFFICE OF RESEARCH AND DEVELOPMENT
U.S. ENVIRONMENTAL PROTECTION AGENCY
ADA, OK 74820

REPRODUCED BY
NATIONAL TECHNICAL
INFORMATION SERVICE
U.S. DEPARTMENT OF COMMERCE
SPRINGFIELD, VA. 22161

TECHNICAL REPORT DATA *(Please read Instructions on the reverse before completing)*		
1. REPORT NO. EPA/600/2-85/065	2.	3. RECIPIENT'S ACCESSION NO. PB85 230170/AS
4. TITLE AND SUBTITLE PLUME2D: TWO-DIMENSIONAL PLUMES IN UNIFORM GROUND WATER FLOW		5. REPORT DATE June 1985 6. PERFORMING ORGANIZATION CODE
7. AUTHOR(S) Jan Wagner, Stephanie Watts, Douglas Kent		8. PERFORMING ORGANIZATION REPORT NO.
9. PERFORMING ORGANIZATION NAME AND ADDRESS Oklahoma State University Stillwater OK 74078		10. PROGRAM ELEMENT NO. ABRD1A 11. ~~CONTRACT/GRANT NO.~~ Coop. Agr. CR-811142
12. SPONSORING AGENCY NAME AND ADDRESS Robert S. Kerr Environmental Research Laboratory Office of Research and Development U.S. Environmental Protection Agency Ada, OK 74820		13. TYPE OF REPORT AND PERIOD COVERED Final 9/83 - 2/85 14. SPONSORING AGENCY CODE EPA/600/15
15. SUPPLEMENTARY NOTES Carl G. Enfield, Project Officer		

16. ABSTRACT

A closed-form analytical solution for two dimensional plumes was incorporated in an interactive computer program. The assumption of an infinite aquifer depth and uniform source mass rate and source location was overcome by using the principal of superposition in space and time. The source code was written in a subset of FORTRAN 77 and can be compiled with FORTRAN IV, FORTRAN 66 as well as FORTRAN 77. As a result, the code is nearly independent of hardware and operating system. The model can be solved for either vertically or horizontally averaged conditions.

17. KEY WORDS AND DOCUMENT ANALYSIS		
a. DESCRIPTORS	b. IDENTIFIERS/OPEN ENDED TERMS	c. COSATI Field/Group
18. DISTRIBUTION STATEMENT RELEASE TO PUBLIC	19. SECURITY CLASS *(This Report)* UNCLASSIFIED 20. SECURITY CLASS *(This page)* UNCLASSIFIED	21. NO. OF PAGES 95 22. PRICE

DISCLAIMER

The information in this document has been funded wholly or in part by the United States Environmental Protection Agency under assistance agreement number CR-811142 to Oklahoma State University. It has been subject to the Agency's peer and administrative review, and it has been approved for publication as an EPA document.

FOREWORD

EPA is charged by Congress to protect the Nation's land, air, and water systems. Under a mandate of national environmental laws focused on air and water quality, solid waste management and the control of toxic substances, pesticides, noise, and radiation, the Agency strives to formulate and implement actions which lead to a compatible balance between human activities and the ability of natural systems to support and nurture life.

The Robert S. Kerr Environmental Research Laboratory is the Agency's center of expertise for investigation of the soil and subsurface environment. Personnel at the Laboratory are responsible for management of research programs to: (a) determine the fate, transport and transformation rates of pollutants in the soil, the unsaturated zone and the saturated zones of the subsurface environment; (b) define the processes to be used in characterizing the soil and subsurface environment as a receptor of pollutants; (c) develop techniques for predicting the effect of pollutants on ground water, soil and indigenous organisms; and (d) define and demonstrate the applicability and limitations of using natural processes, indigenous to the soil and subsurface environment, for the protection of this resource.

This project was initiated to develop an interactive computer model which could be utilized to predict toxic chemical fate in homogeneous aquifers. The model should be useful in making comparisons, between chemicals, for idealized homogeneous aquifers. This model is not intended for addressing site specific problems where there is significant heterogeneity in the aquifer.

Clinton W Hall

Clinton W. Hall
Director
Robert S. Kerr Environmental
Research Laboratory

ABSTRACT

A closed-form analytical solution for two dimensional plumes was incorporated in an interactive computer program. The assumption of an infinite aquifer depth and uniform source mass rate and source location was overcome by using the principal of superposition in space and time. The source code was written in a subset of FORTRAN 77 and can be compiled with FORTRAN IV, FORTRAN 66 as well as FORTRAN 77. As a result, the code is nearly independent of hardware and operating system. The model can be solved for either vertically or horizontally averaged conditions.

TABLE OF CONTENTS

INTRODUCTION.....1

SECTION I - MATHEMATICAL DEVELOPMENT.....2
Vertically-averaged solution.....7
Horizontally-averaged solution.....13
Assumptions and Limitations.....17
Superposition.....17

SECTION II - COMPUTER PROGRAM.....25
Basic Input Data.....25
Edit Commands.....30

SECTION III - APPLICATIONS.....32

REFERENCES.....40

APPENDIX A - Example Problems.....A-1

APPENDIX B - Description of Program PLUME2D.....B-1

APPENDIX C - Numerical Analysis of the Hantush Well Function.....C-1

APPENDIX D - Listing of Program PLUME2D.....D-1

APPENDIX E - Listing of Utility Function Subroutines.....E-1

LIST OF FIGURES

Figure 1 - Coordinate system for vertically averaged solution.....6

Figure 2 - Coordinate system for horizontally averaged solution.....6

Figure 3 - Decomposition of a variables source rate using superposition in time.....20

Figure 4 - Use of image sources to account for aquifers of finite depth.....22

Figure 5 - Superposition in space to account for barrier boundaries.....24

Figure 6 - Results of hexavalent chromium plume simulation at 3280 days..33

Figure 7 - Results of hexavalent chromium spill simulation at 365 days...36

LIST OF TABLES

Table 1 - Edit Commands.....31

Table 2 - Comparison of Concentrations Calculated Using Superposition in Time and an Analytical Solution for an Instantaneous Line Source.....38

Table 3 - Typical Values for Aquifer Properties 39

INTRODUCTION

Relatively simple analytical methods can often be used to evaluate ground-water contamination problems, depending upon the complexity of the system and the availability of field data. Analytical models can also serve as valuable tools in developing parameters for more sophisticated numerical models. Although the numerical evaluation of an analytical solution to a ground-water problem may be mathematically complex, analytical models are well suited for interactive use on digital computers. Many analytical solutions to ground-water contamination problems can be coded on programmable hand-held calculators. In general, very few input parameters are required to define a given problem and numerical results can be calculated in a few seconds.

This report presents analytical solutions to two ground-water pollution problems -- two-dimensional plumes in uniform ground-water flow. An interactive computer code has been developed which enables the user to modify the definition of a given problem, and thus gain some insight into the effects of various parameters on the extent of a contaminant plume.

SECTION I

MATHEMATICAL DEVELOPMENT

The differential equation describing the conservation of mass of a component in a saturated, homogeneous aquifer with uniform, steady flow in the x-direction can be written as

$$\frac{\partial C_T}{\partial t} + V \frac{\partial (C)}{\partial x} = D_x \frac{\partial^2 (\theta C)}{\partial x^2} + D_y \frac{\partial^2 (\theta C)}{\partial y^2} + D_z \frac{\partial^2 (\theta C)}{\partial z^2} - r_t \qquad (1)$$

where

C	= component mass per unit of fluid phase	M/L^3
C_T	= total component mass per unit volume of aquifer	M/L^3
D_x	= dispersion coefficient in x-direction	L^2/t
D_y	= dispersion coefficient in y-direction	L^2/t
D_z	= dispersion coefficient in z-direction	L^2/t
r_t	= rate of degradation of mass per unit volume of aquifer	M/L^3t
V	= Darcy, or seepage, velocity in the x-direction	L/t
$x,y,z,$	= rectangular coordinates at the point of interest	L
θ	= porosity of porous media	L^3/L^3

The total mass of a component per unit volume of aquifer is distributed as dissolved solute in the fluid phase and adsorbed solute on the solid matrix. Let

C_s = component mass per unit mass of solid M/M

and

ρ_B = bulk density of the aquifer, or the mass of solids per unit volume of the aquifer M/L^3.

The total component mass per unit volume of aquifer can be expressed as

$$\frac{\text{Mass}}{\text{Unit Volume}} = \frac{\text{Volume of voids}}{\text{Unit Volume of aquifer}} \frac{\text{Component Mass}}{\text{Volume of voids}} + \frac{\text{Mass of solids}}{\text{Unit volume of aquifer}} \frac{\text{Component Mass}}{\text{Mass of solids}}$$

or

$$C_T = \theta C + \rho_B C_s \tag{2}$$

and, the rate of accumulation of mass in the aquifer becomes

$$\frac{\partial C_T}{\partial t} = \theta \frac{\partial C}{\partial t} + \rho_B \frac{\partial C_s}{\partial t} \tag{3}$$

In general, $C_s = f(C)$ and

$$\frac{\partial C_s}{\partial t} = \frac{dC_s}{dC} \frac{\partial C}{\partial t} \tag{4}$$

For a linear equilibrium adsorption isotherm,

$$\frac{dC_s}{dC} = K_d \frac{M/M}{M/L^3} \tag{5}$$

where K_d is a distribution constant.

The change in concentration per unit volume of porous media, $\partial C_T/\partial t$, can be written in terms of fluid phase concentration, C, by substituting Equations 4 and 5 into Equation 3. Therefore,

$$\frac{\partial C_T}{\partial t} = \theta \frac{\partial C}{\partial t} + \rho_B K_d \frac{\partial C}{\partial t}$$

or

$$\frac{\partial C_T}{\partial t} + (\theta + \rho_B K_d) \frac{\partial C}{\partial t} \qquad (6)$$

The rate of degradation of component mass per unit volume of porous media is also distributed between the solid and liquid phases, or

$$\frac{\text{Rate of mass degraded}}{\text{Unit volume of aquifer}} = \frac{\text{Rate of mass degraded}}{\text{Unit volume of fluid}} \frac{\text{Volume of fluid}}{\text{Volume of aquifer}}$$

$$+ \frac{\text{Rate of mass degraded}}{\text{Unit mass of solid}} \frac{\text{Mass of solid}}{\text{Volume of aquifer}}$$

Now, the rate of change in total mass per unit volume of aquifer due to reaction can be written as

$$r_t = \frac{\partial C_T}{\partial t} = \theta \frac{\partial C}{\partial t} + \rho_B \frac{\partial C_s}{\partial t} \qquad (7)$$

The concentration on the solid, C_s, is related to the concentration in the liquid, C, through the linear adsorption isotherm assumed previously, and

$$r_t = \frac{\partial C_T}{\partial t} = (\theta + \rho_B K_d) \frac{\partial C}{\partial t} \qquad (8)$$

Assuming first order decay kinetics, the rate of decrease in fluid phase and solid phase concentrations due to reaction can be expressed as

$$\frac{\partial C}{\partial t} = \lambda C \tag{9}$$

and

$$\frac{\partial C_s}{\partial t} = \lambda C_s$$

respectively, where λ is a rate constant (1/t), and

$$r_t = (\theta + \rho_B K_d)\lambda C \tag{10}$$

Equation 1 can now be written in terms of the fluid concentration. Substituting Equations 6 and 10 and recalling that for a homogeneous porous medium the porosity, θ, is constant, Equation 1 becomes

$$(1 + \frac{\rho_B}{\theta} K_d) \frac{\partial C}{\partial t} + V^* \frac{\partial C}{\partial x} = D_x \frac{\partial^2 C}{\partial x^2} + D_y \frac{\partial^2 C}{\partial y^2} + D_x \frac{\partial^2 C}{\partial z^2} - (1 + \frac{\rho_B}{\theta} K_d) \lambda C \tag{11}$$

where $V^* \equiv \frac{V}{\theta}$ is the average interstitial, or pore, velocity. Defining a "retardation coefficient" as

$$R_d = 1 + \frac{\rho_B}{\theta} K_d \tag{12}$$

the differential equation describing the conservation of mass in the aquifer becomes

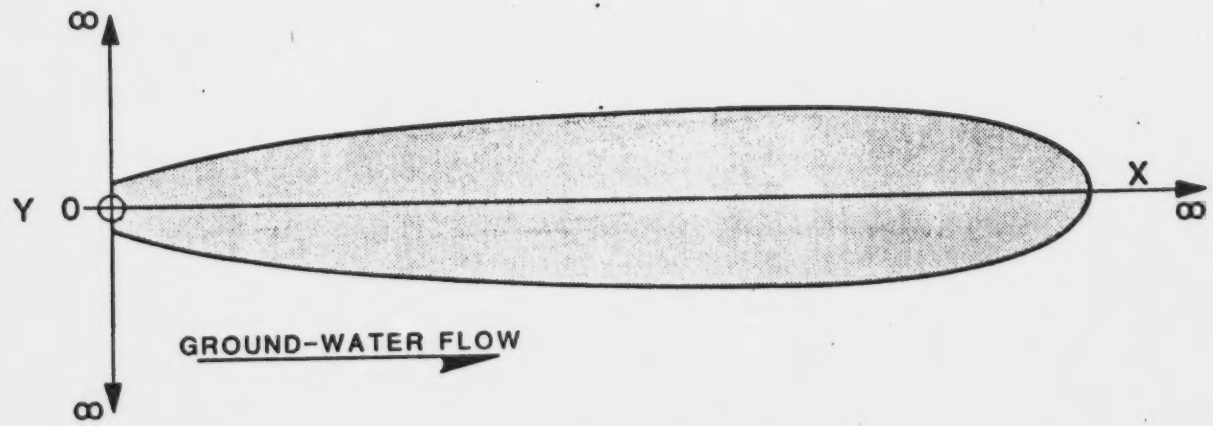


Figure 1. Coordinate system for vertically averaged solution.

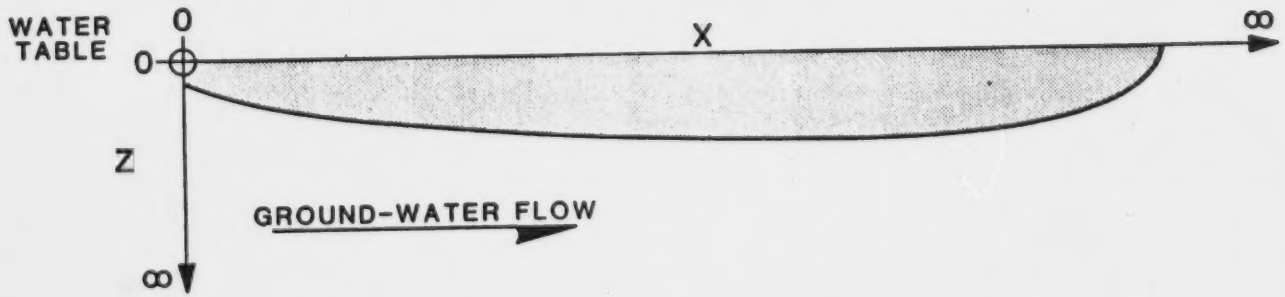


Figure 2. Coordinate system for horizontally averaged solution.

$$R_d \frac{\partial C}{\partial t} + V^* \frac{\partial C}{\partial x} = D_x \frac{\partial^2 C}{\partial x^2} + D_y \frac{\partial^2 C}{\partial y^2} + D_z \frac{\partial^2 C}{\partial z^2} - R_d \lambda C \qquad (13)$$

Equation 13 is a linear partial differential equation which can be integrated analytically to yield an expression for concentration as a function of time and position.

Solutions of Equation 13 for two types of ground-water contamination problems are presented in the following paragraphs. The first is a vertically-averaged solution which describes a contaminant plume in the x-y plane (Figure 1). The second is a horizontally-averaged solution, describing a contaminant plume in the x-z plane (Figure 2).

Vertically-averaged solution. The vertically-averaged solution applies to a homogeneous aquifer of infinite aerial extent and finite depth. The contaminant is assumed to be well mixed over the saturated thickness. The source of contaminant is a vertical line source located at the origin of a coordinate system in the x-y plane. This conceptual model would apply to an injection well which fully penetates the saturated zone.

Wilson and Miller (1978) have also applied this solution downstream from a contaminant source at the surface of the water table. For a relatively thin saturated zone, vertical dispersion will result in mixing vertically. The concentration distribution can be considered as being two-dimensional in a horizontal plane at distances downstream of the source sufficient for the concentration distribution to become uniform with depth. Mathematically, the problem is treated as an infinite aquifer with a line source at the origin. The vertically-averaged formulation of Equation 13 is

$$R_d \frac{\partial C}{\partial t} + V^* \frac{\partial C}{\partial x} = D_x \frac{\partial^2 C}{\partial x^2} + D_y \frac{\partial^2 C}{\partial y^2} - R_d \lambda C \qquad (14)$$

The boundary conditions can be stated mathematically as follows

$$C(x,y,0) = 0 \tag{15a}$$

$$C(x,\pm\infty,t) = 0 \tag{15b}$$

$$C(\pm\infty,y,t) = 0 \tag{15c}$$

A solution to Equation 14 with the Equation 15 boundary conditions and a continuous source of strength C_oQ' can be written as (Hunt, 1978; Wilson and Miller, 1978):

$$C = \frac{C_oQ' \text{ EXP } \left(\frac{V^*x}{2D_x}\right)}{4\pi\theta \left(D_xD_y\right)^{0.5}} W(U,B) \tag{16}$$

where

$$U = \frac{\left(\frac{Vx}{D_x}\right)^2 + \frac{D_x}{D_y}\left(\frac{V^*y}{D_x}\right)^2}{\frac{4\ V^{*2}\ t}{R_d\ D_x}} \tag{17}$$

and

$$B = \frac{1}{2}\left[\left(\frac{V^*x}{D_x}\right)^2 + \frac{D_x}{D_y}\left(\frac{V^*y}{D_x}\right)^2\right]^{1/2}\left[1 + \frac{4D_xR_d\lambda}{V^{*2}}\right]^{1/2} \tag{18}$$

and C_oQ' (M/t/L) is the contaminant source rate per unit depth of the saturated zone.

The function W(U,B) is defined as

$$W(U,B) = \int_U^\infty \frac{1}{\xi} \text{EXP}\ (-\xi - \frac{B^2}{4\xi})\ d\xi \qquad (19)$$

where ξ is a dummy integration variable. This function is often referred to as the "well function for leaky artesian aquifers" (Hantush; 1956, 1964).

The steady-state solution of Equations 14 and 15 can be obtained by noting as $t \to \infty$, $U \to 0$ and the well function (Hantush, 1956) can be expressed as

$$W(0,B) = 2K_o(B) \qquad (20)$$

where $K_o(B)$ is the modified Bessel function of the second kind of order zero. At steady-state the vertically-averaged solution can be written as

$$C = \frac{C_oQ'\ \text{EXP}\ (\frac{V^*x}{2D_x})}{2\pi\theta\ (D_xD_y)^{0.5}}\ K_o(B) \qquad (21)$$

The units of the variables in Equations 16 and 21 can be eliminated by defining the following dimensionless groups:

$$\text{Modified Peclet Numbers} \sim \frac{\text{Convective mass transport}}{\text{Dispersive mass transport}}$$

$$Pe_x = \frac{V^* x}{D_x} \tag{22}$$

$$Pe_y = \frac{V^* y}{D_x} \tag{23}$$

Damkohler Group II $\sim \frac{\text{Mass decay rate}}{\text{Mass dispersion rate}}$

$$D_k = \frac{D_x R_d \lambda}{V^{*2}} \tag{24}$$

Number of Pore Volumes Injected $\sim \frac{\text{Mass transport rate}}{\text{Mass accumulation rate}}$

$$I = \frac{V^{*2} t}{R_d D_x} \tag{25}$$

Dimensionless Source Term $\sim \frac{\text{Mass injection rate}}{\text{Mass diffusion rate}}$

$$\Gamma = \frac{Q'}{\theta (D_x D_y) 0.5} \tag{26}$$

Dimensionless Concentration

$$\gamma = \frac{C}{C_o} \tag{27}$$

Note that the number of pore volumes injected can be written as

$$I = \frac{V^2 t}{D_x R_d} = \left(\frac{V^* L}{D_x}\right)^2 \quad \frac{D_x t}{R_d L^2} \tag{28}$$

where L is a characteristic length defined as

$$L^2 = x^2 + \frac{D_x}{D_y} y^2 \tag{29}$$

The first group on the right-hand-side of Equation 28 is the Modified Peclet number

$$Pe_{xy} = \left[\left(\frac{V^* x}{D_x}\right)^2 + \quad \frac{D_x}{D_y} \quad \left(\frac{V^* y}{D_x}\right)^2\right]^{1/2}$$

or

$$Pe_{xy} = (Pe_x^2 + \beta\, Pe_y^2)^{1/2} \tag{30}$$

where

$$\beta = \frac{D_x}{D_y}$$

The second group on the right-hand-side of Equation 28 is a dimensionless time variable,

$$\tau = \frac{D_x t}{R_d L^2}. \tag{31}$$

The transient and steady-state solutions to Equations 14 which are given by Equations 16 and 21 can be written in terms of the dimensionless variables defined above. The transient solution is

$$\gamma = \frac{\Gamma}{4\pi} \text{EXP}\left(\frac{1}{2} Pe_x\right) W(U,B) \tag{32}$$

and at steady state

$$\gamma = \frac{\Gamma}{2\pi} \text{EXP}\left(\frac{1}{2} Pe_x\right) K_o(B) \tag{33}$$

with

$$U = \frac{Pe_{xy}^{\ 2}}{4I} \tag{34}$$

and

$$B = \frac{1}{2} Pe_{xy}(1 + 4D_k)^{\frac{1}{2}} \tag{35}$$

The values of dimensionless concentrations evaluated using Equation 32 or Equation 33 are valid for any consistent set of units. Using dimensionless variables also tends to "scale" numerical values when working in various systems of units.

Horizontally-averaged solution. Consider a homogeneous aquifer with a continuous line source of infinite length located at the water table and normal to the direction of ground-water flow as shown in Figure 2. In other words the tracer is assumed to be well mixed over the width of the aquifer. A problem which might fit this conceptual model is seepage from a trench perpendicular to the direction of ground-water flow.

The horizontally-averaged formulation of Equation 13 is

$$R_d \frac{\partial C}{\partial t} + V^* \frac{\partial C}{\partial x} = D_x \frac{\partial^2 C}{\partial x^2} + D_z \frac{\partial^2 C}{\partial z^2} - R_d \lambda C \tag{36}$$

For an aquifer of infinite depth and a uniform continuous line source, the appropriate boundary conditions can be written as follows

$$C(x,z,0) = 0 \tag{37a}$$

$$C(\pm\infty,z,t) = 0 \tag{37b}$$

$$C(x,\infty,t) = 0 \tag{37c}$$

$$\frac{\partial C(x,0,t)}{\partial z} = 0 \tag{37d}$$

A solution to Equation 36 with the Equation 37 boundary conditions and a continuous line source of strength C_oQ' is

$$C = \frac{C_oQ' \ EXP(\frac{V^*x}{2D_x})}{2\pi\theta \ (D_xD_z)^{0.5}} \ W(U,B) \tag{38}$$

At steady state the horizontally-averaged solution can be written as

$$C = \frac{C_oQ' \ EXP \ (\frac{V^*x}{2D_x})}{\pi\theta \ (D_xD_z)^{0.5}} \ K_o(B) \tag{39}$$

where

$$U = \frac{\left(\frac{V^*_x}{D_x}\right)^2 + \frac{D_x}{D_z}\left(\frac{V^*_z}{D_x}\right)^2}{\frac{4\,V^{*2}\,t}{R_d\,D_x}} \tag{40}$$

$$B = \frac{1}{2}\left[\left(\frac{V^*_x}{D_x}\right)^2 + \frac{D_x}{D_z}\left(\frac{V^*_z}{D_x}\right)^2\right]^{1/2}\left[1 + \frac{4D_x R_d \lambda}{V^{*2}}\right]^{1/2} \tag{41}$$

and Q' ($L^3/t/L$) is the volumetric contaminant source rate per unit width of the aquifer (or unit length of the line source).

Changing subscripts, the definition of the dimensionless groups leads to

$$Pe_z = \frac{V^*_z}{D_x} \tag{42}$$

and

$$I = \frac{Q'}{\theta(D_x D_z)0.5} \tag{43}$$

with

$$Pe_{xz} = (Pe_x^2 + \beta\, Pe_z^2)^{1/2} \tag{44}$$

where

$$\beta = \frac{D_x}{D_z} \tag{45}$$

By substituting the dimensionless groups described in vertically-averaged solution and those defined above, Equations 38 through 41 can be written in terms of dimensionless variables.

The transient solution becomes

$$\gamma = \frac{\Gamma}{2\pi} \text{EXP}\left(\frac{1}{2} Pe_x\right) W(U,B) \tag{46}$$

and at steady state, the horizontally-averaged solution is

$$\gamma = \frac{\Gamma}{\pi} \text{EXP}\left(\frac{1}{2} Pe_x\right) K_o(B) \tag{47}$$

where

$$U = \frac{Pe_{xz}^{\ 2}}{4I} \tag{48}$$

and

$$B = \frac{1}{2} Pe_{xz}(1 + 4D_k)^{\frac{1}{2}} \tag{49}$$

The similarity of the solutions of the vertically-averaged and horizontally-averaged problems facilitates their numerical evaluation using a common computational algorithm. For the same numerical values of the independent variables, concentration values for the horizontally-averaged solutions are obtained by doubling the vertically-averaged solution values.

Assumptions and Limitations

Equations 32-33 and 46-47 can be used to calculate the concentrations in leachate plumes under the following assumptions and limitations:

1. The ground-water flow regime is completely saturated.
2. All aquifer properties are constant and uniform throughout the aquifer.
3. All ground-water flow is horizontal, continuous, and uniform throughout the aquifer.
4. The aquifer is infinite in extent for the vertically-averaged solution, or semi-finite in extent for the horizontally-averaged solution.
5. The leachate source is a line located at the origin of the coordinate system.
6. The mass flow rate of the source is constant.
7. At zero time the concentration of leachate in the aquifer is zero.

The assumptions of an infinite aquifer depth and a uniform source mass rate can be overcome by using the principles of superposition in space and time, respectively (Walton, 1962). Both of these provisions have been incorporated in the computer program described in the next section.

Superposition

The differential equation describing component mass concentration in a porous medium, Equation 1, is a linear partial differential equation. The principal of superposition can be used directly to solve complex ground-water contamination problems in terms of the simplier solutions described above. Unfortunately, the scattered applications of this principle are not explained in any single reference. Some texts indicate that superposition means that any sum of solutions is also a solution. Superposition is commonly used to generate a linear no-flow boundary condition through the use of "image wells"

or to simulate multiple sources and sinks (Walton; 1962, 1970). The principle of superposition is also complicated by referring to the "Duhamel theorem," the "Faltung integral," and/or "convolution integrals." These terms often have no apparent physical interpretation. For the purposes of this report, "superposition in space" will refer to the approximation of sources of finite area or volume as the sum of a finite number of point sources or the generation of no-flow boundaries using image wells. "Superposition in time" will refer to the approximation of a variable source rate of contamination as the sum of a finite number of constant source rates distributed in time.

Both the horizontally-averaged and vertically-averaged solutions can be used to simulate aquifers of finite width or depth, respectively, or plane sources of finite width. Applications of this type require a thorough understanding of the physical interpretation of the principal of superposition.

Some applications are relatively straight forward, and the computer program provides for the approximation of a non-uniform source rate using superposition in time. Multiple sources and aquifers of finite thickness are also included using superposition in space.

Consider the variable source of contamination shown in Figure 3. The solutions of the governing differential equation presented in this report are of the form

$$C(x,z,t) = C_o Q' \, f(x,z,t) = \dot{Q}' \, f(x,z,t) \qquad (50)$$

where $\dot{Q}'$ is the source mass rate per unit length. The principle of superposition in time can be written for any position as

$$C(x,z,t) = \sum_{i=1}^{n} \dot{Q}'_i \; f(x,z,t_i) \tag{51}$$

Now, the variable rate schedule shown in Figure 3a can be decomposed into a series of positive and negative mass rates as shown in Figure 3b. The concentration at a point x,y,z at the end of the simulation, t_s, can be evaluated as

$$\begin{aligned} C(x,y,z,t) = {} & \dot{Q}'_1 \; f(x,y,z,t_1) - \dot{Q}'_1 \; f(x,y,z,t_2) \\ & + \dot{Q}'_2 \; f(x,y,z,t_2) - \dot{Q}'_2 \; f(x,y,z,t_3) \\ & + \dot{Q}'_3 \; f(x,y,z,t_3) - \dot{Q}'_3 \; f(x,y,z,t_4) \\ & + \dot{Q}'_4 \; f(x,y,z,t_4) \end{aligned} \tag{52}$$

In general terms

$$C(x,y,z,t_s) = \sum_{i=1}^{n} \; (\dot{Q}'_i - \dot{Q}'_{i-1}) \; \; f(x,y,z,t_i) \tag{53}$$

with $\dot{Q}'_0 = 0$

Note the time corresponding to a given source rate, t_i, is the period beginning with the start of the given rate to the end of the simulation period; time is not the duration of a given rate. For ease of application, Equation 53 can be rewritten as

$$C(x,y,z,t_s) = \sum_{k=1}^{n} \; (\dot{Q}'_k - \dot{Q}'_{k-1}) \; f(x,y,z,t_s - t_{k-1}) \tag{54}$$

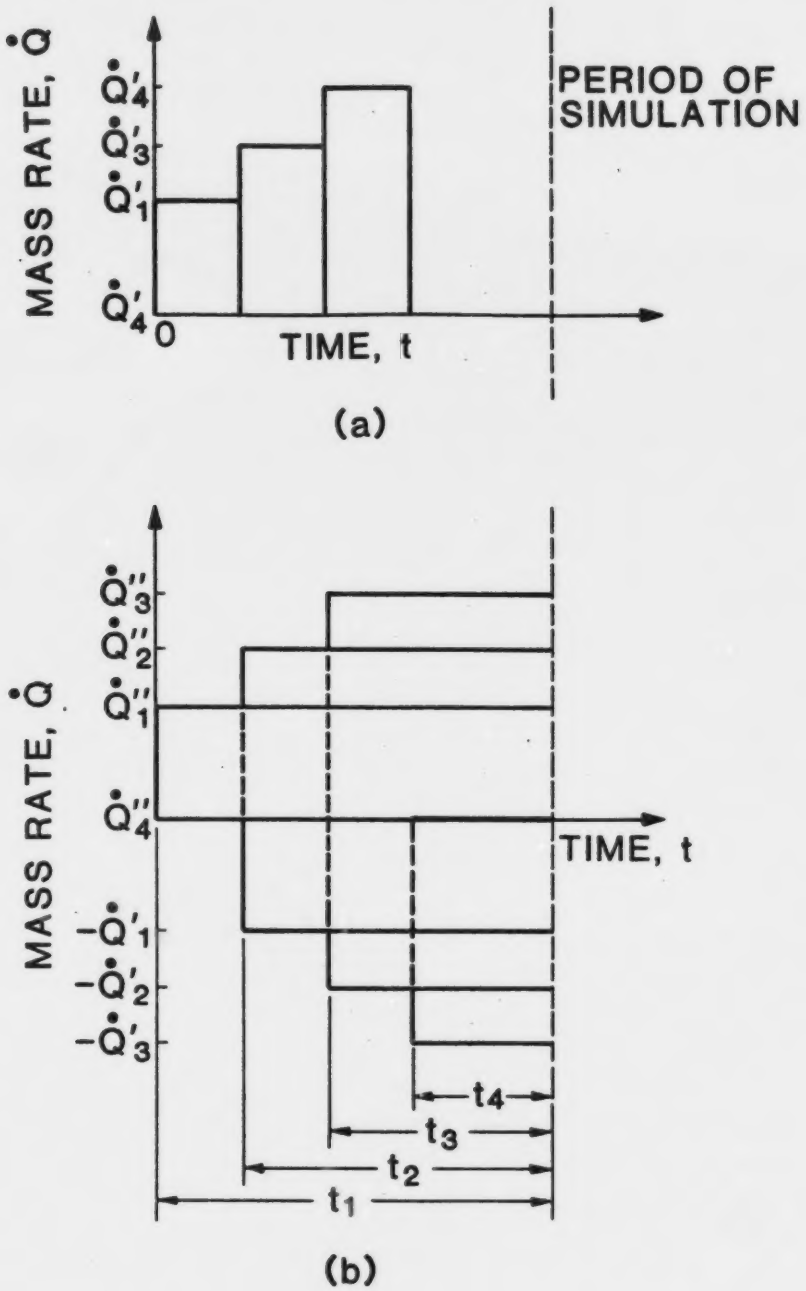


Figure 3. Decomposition of a variable source rate using superposition in time.

where t_{k-1} is the time corresponding to the end of mass rate Q_{k-1} or the beginning of rate Q_k with $Q_o = 0$ and $t_o = 0$.

A continuous non-uniform rate schedule may be approximated as closely as desired by increasing the number of discrete rates in the source rate schedule. In theory an infinite number of discrete rates would be required. An understanding of the physical problem and the assumptions incorporated in the mathematical model are the best guidelines for decomposing a continuous non-uniform source of contamination.

The influence of geohydrologic boundaries on the movement of a tracer is similar to the influence of these boundaries on the drawdown response of an aquifer to pumping. The applications of image well theory described by Walton (1962, 1970) can be extended to the horizontally-averaged solution to the solute transport problem considered in this report. The following discussion parallels Walton's examples of the use of image wells to account for barrier boundaries.

Consider the contaminant plume which would exist if the aquifer were of infinite depth as shown in Figure 4a. If the contaminant plume was to intersect an impermeable base of the aquifer as shown in Figure 4b, the vertical concentration gradient must change since there can be no transport of mass across the boundary as a result of dispersion. In mathematical terms

$$D_z \frac{\partial C}{\partial z} = 0$$

at $z = B$. Now, if an imaginary, or image, source were placed across the boundary at a distance equal to the depth of the aquifer, as shown in Figure 4c, this source would create a concentration gradient from the boundary to the image water table equal to the concentration gradient from the boundary

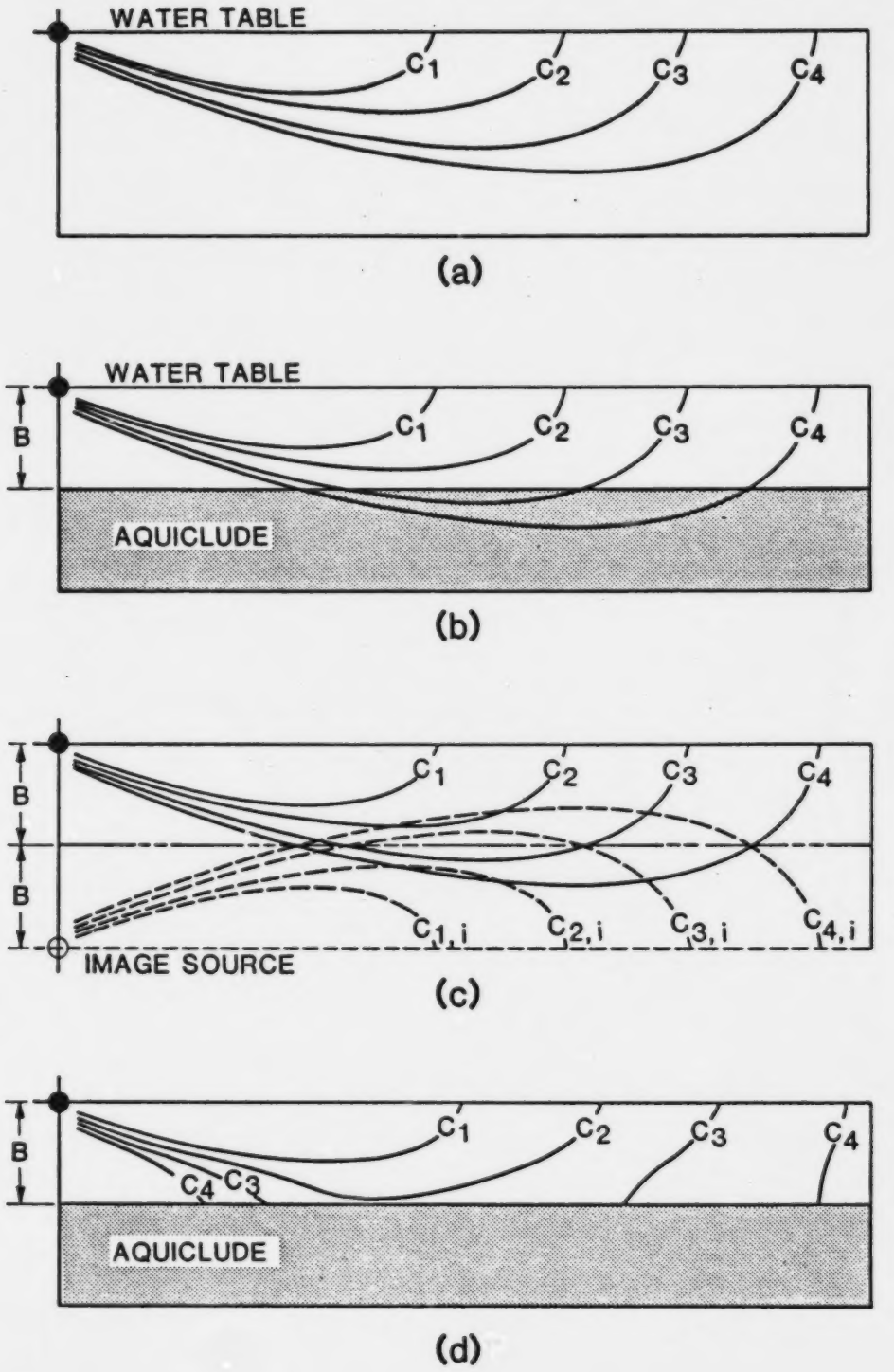
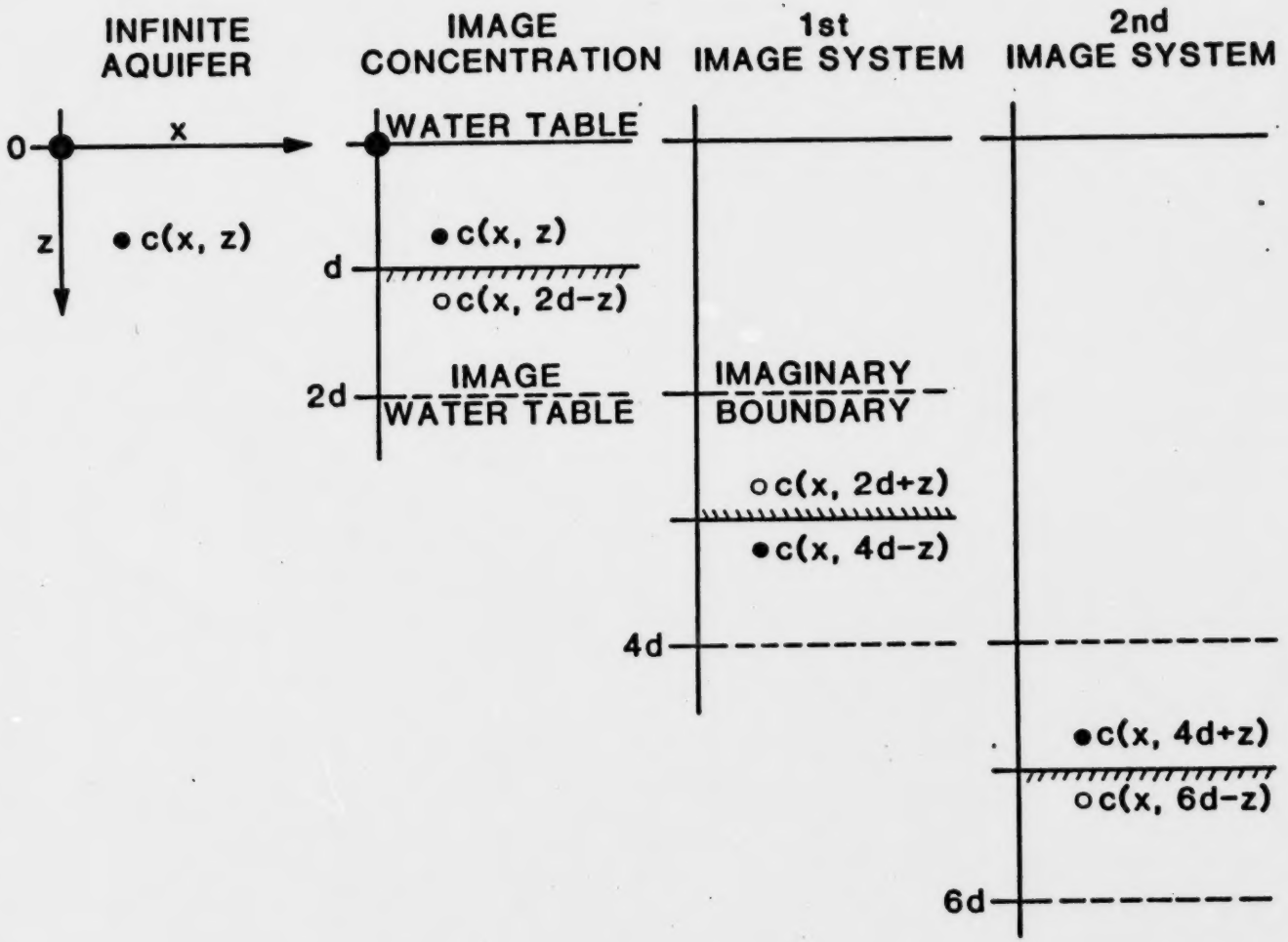


Figure 4. Use of image sources to account for aquifers of finite depth.

to the real water table. A "concentration divide" would be established at boundary, and the no-transport boundary condition ($\partial C/\partial z = 0$) would be satisfied.

The imaginary system of a contaminant source and its image in an aquifer of infinite depth satisfies the boundary conditions dictated by the finite depth system. The resultant concentration distribution is the sum of concentrations in both the real and image systems as shown in Figure 4d.

In theory an infinite number of image systems may be required. For example, if the plume in the infinite system intersects the water table in the image system, a second no-transport boundary is encountered as shown in Figure 5. This boundary can be handled by introducing another image system across the imaginary boundary and equidistant from the first image system. This process of adding image systems could be repeated indefinitely. In practice only a few image systems are required. The computer program automatically introduces an appropriate number of image systems.



$$c'(x, z) = c(x, z) + \sum_{n=1}^{\infty} \left[c(x, 2nd - z) + c(x, 2nd + z) \right]$$

Figure 5. Superposition in space to account for barrier boundaries.

SECTION II

COMPUTER PROGRAM

The computer program evaluates the analytical solutions of the differential equation describing concentration distributions in two-dimensional plumes with uniform ground-water flow. The program has been designed for interactive use and requires input data under two modes of operation - "Basic Input Data" and "Edit."

Basic Input Data

Basic input data are required to initiate a new problem using the PLUME2D program. The user is prompted for the required data through a series of input commands described below. Numeric data may be entered through the keyboard with or without decimal points and multiple data entries should be separated by comma(s). The first basic input command is:

```
ENTER TITLE
?
```

Any valid keyboard characters can be used. The first 60 characters will be retained for further problem identification.

The second input command is used to select the vertically-averaged solution or the horizontally-averaged solution. The command is:

```
ENTER COORDINATE SYSTEM
XY FOR VERTICALLY-AVERAGED SOLUTION
XZ FOR HORIZONTALLY-AVERAGED SOLUTION
?
```

Either of the indicated responses is valid.

The next three input commands define the units for all variables used in the calculations. Any consistent set of units may be used.

```
ENTER UNITS FOR LENGTH (2 CHARACTERS)
?
```

Any valid keyboard characters can be used. The first two characters will be retained for identifying the units of the length dimensions which may be required for other input data or output listings.

ENTER UNITS FOR TIME (2 CHARACTERS)
?

Any valid keyboard characters can be used. The first two characters will be retained for identifying the units of the time dimensions which may be required for other input data or output listings.

ENTER UNITS FOR CONCENTRATION (6 CHARACTERS)
?

The first six characters of any valid keyboard entries will be retained for identifying the concentration units for data input and output.

The remaining input commands are used to initialize all variables for a given problem. They include both aquifer and contaminant parameters. Input data errors which may interrupt the computational sequence are detected by the program and a command is issued to reenter the data for the appropriate variable.

ENTER SATURATED THICKNESS, (0 FOR INFINITE THICKNESS), L
?

If horizontally-averaged solution was selected (x-z coordinate system) this request is issued. The saturated thickness must be entered in the units requested with dimensions of L. If a zero or negative value is entered, the calculations will be carried out assuming an aquifer of infinite depth. The program automatically includes up to 20 image wells for aquifers of finite depth.

ENTER AQUIFER POROSITY
?

Enter the volume void fraction.

ENTER SEEPAGE VELOCITY, L/t
?

The seepage, or interstitial, velocity must be entered with dimensions of L/t in the units requested. Numerical values must be greater than zero.

```
ENTER RETARDATION COEFFICIENT
?
```

The retardation coefficient includes the effects of absorption of the tracer on the solid matrix (see Section I for discussion). The numerical value must be greater than 1.0, or equal to 1.0 if absorption is neglected.

```
ENTER X DISPERSION COEFFICIENT, SQ L/t
?
```

Dispersion coefficients have dimensions of L^2/t and must be entered in the units requested. Numerical values must be greater than zero.

If the X-Y coordinate system has been selected, the next command is:

```
ENTER Y DISPERSION COEFFICIENT, SQ L/t
?
```

If, instead, the X-Z coordinate system has been selected, a command for the Z dispersion coefficient will be issued.

```
ENTER Z DISPERSION COEFFICIENT, SQ L/t
?
```

The subsequent command will be:

```
ENTER DECAY CONSTANT, 1/t
?
```

The first order decay constant has dimensions of 1/t and must be entered in the units requested. The decay constant must be greater than, or equal to, zero.

```
SELECT TRANSIENT OR STEADY-STATE SOLUTION
  TR FOR TRANSIENT SOLUTION
  SS FOR STEADY-STATE SOLUTION
?
```

Selection of the transient solution also allows the approximation of a nonuniform rate schedule by a series of uniform rates (see Section I for discussion). Approximation is accomplished through superposition of a series

of uniform rates. If steady-state solution is chosen, the steady state concentration will be evaluated.

ENTER THE NUMBER OF SOURCES (MAXIMUM OF N)
?

The number of sources of contaminant should be entered. The value entered must be greater than zero.

MASS RATES HAVE UNITS OF (M/L^3) (L^3/t)
TIME HAS UNITS OF t

This statement reminds the user of the units that will be used for mass rates and for time. All mass-rate and time values entered must be in these units.

The next series of commands will be repeated for each source.

ENTER X and Z COORDINATES OF SOURCE I (L)
?

The input units for the coordinates must be in the units requested. The Z-coordinate must be greater than or equal to zero. If, instead, the X-Y coordinate system has been selected, the following command is issued:

ENTER X AND Y COORDINATES OF SOURCE I (L)
?

If the transient solution was chosen the following two commands will be issued.

ENTER THE NUMBER OF RATES FOR SOURCE I (MAXIMUM OF N)
?

The number of uniform rates used to approximate a nonuniform rate schedule for this source is entered. The value must be greater than zero.

SOURCE I, RATE J STARTS AT TIME t
ENTER MASS RATE AND ENDING TIME
?

The source mass rate is entered in units of concentration times the volumetric rate. Note the actual source concentration and rate are not required, but the units must be consistent. The time units must also be consistent.

If the steady-state solution has been selected, the following command will be entered instead of the two previously listed commands.

ENTER STEADY-STATE MASS RATE I
?

The next two basic input commands are used to define the matrix of observation points, or coordinates at which concentration will be evaluated.

ENTER XFIRST, XLAST, DELTAX (L)
?,?,?

The input units for the coordinates must also be in the units requested. XFIRST and XLAST can be positive or negative values. A zero entry for DELTAX will result in a single X-coordinate observation. Results of calculations for multiple X-coordinates will be listed from XFIRST to XLAST.

ENTER YFIRST, YLAST, DELTAY (L)
?,?,?

Any of the numerical values used to define the Y-coordinates of observation points may be positive or negative. If the X-Z coordinate system has been selected, a command to enter the Z-coordinates, rather than the Y-coordinates, will be issued.

ENTER ZFIRST, ZLAST, DELTAZ (L)
?,?,?

ENTER TFIRST, TLAST, DELTAT (t)
?,?,?

The beginning value and ending value of the time interval of contaminant transport being modeled is entered. Both TFIRST and TLAST must be positive values in the units requested. A zero entry for DELTAT will result in model output at a single value of time.

Edit Commands

Once the basic input data have been entered, the problem as currently defined is listed and the program enters the "edit" mode. The edit commands are listed in Table 1 and are also listed the first time the program enters the edit mode. The request for information is

ENTER NEXT COMMAND?

One of the reponses from Table 2 should be given. If the response is incorrect or improperly formulated the statement

ERROR IN LAST COMMAND -- REENTER?

is issued. Error messages for invalid numerical data will be issued as described under the Basic Input Commands. The request for information will be repeated until one of the responses MU, LI, RN, NP, or DN is entered.

MU will list the table of edit commands.

LI will list the problem as currently defined.

RN will initiate the calculation of concentrations and print the results.

NP will request a complete new problem using the "Basic Input Data" dialog.

DN will terminate the program.

A listing of the dialog and the results for the example problem discussed in Section III are included in Appendix A.

Although many tests for valid input data and properly formulated edit commands have been embedded in the program, the user is encouraged to correct "keyboard errors" before the data are transmitted. These precautions will serve to minimize the frustration of program termination as a result of fatal errors during execution of the numerical computations.

Table 1

EDIT COMMANDS

Command	Variable changed/Execution
ST	Saturated Thickness
PO	Porosity
VX	New Seepage Velocity
RD	Retardation Coefficient
DE	Decay Constant
DX	X-Dispersion Coefficient
DY	Y-Dispersion Coefficient
DZ	Z-Dispersion Coefficient
RT	Source Rate Schedule
OB	Observation Points
XC	X-Coordinates
YC	Y-Coordinates
ZC	Z-Coordinates
TC	Observation Times
CS	Change Solution/Sources
MU	Menu of Edit Commands
LI	List input data
RN	Run
NP	New Problem
DN	Done

SECTION III

APPLICATIONS

The case history of ground-water contamination with hexavalent chromium in South Farmingdale, Nassau County, New York, (Perlmutter and Lieber, 1970), has been used as an example of the application of the two-dimensional plume model. The contaminant plume has been modeled numerically by Pinder (1973) and analytically by Wilson and Miller (1978). Details of the hydrologic system are described in the above references. A brief summary of the problem is presented in the following paragraphs.

The aquifer is assumed to have a saturated thickness of 33.52 m with a porosity of 0.35. Perlmutter and Lieber (1970) estimated the average seepage velocity to be approximately 0.366 m/dy. Using Pinder's (1973) values of dispersivity, α_x = 21.3 m and α_y = 4.27 m, and x and y dispersion coefficients are

$$D_x = (21.3 \text{ m})\ (0.366 \text{ m/dy}) = 7.70 \text{ m}^2/\text{dy}$$

and

$$D_y = (4.27 \text{ m})\ (0.366 \text{ m/dy}) = 1.56 \text{ m}^2/\text{dy}$$

The source of contamination consisted of three metal-plating-waste disposal ponds as shown in Figure 6. The mass rate of chromium entering the aquifer has been estimated at 23.6 kg/dy during the nine year period from 1941 through 1949 (Perlmutter and Lieber, 1970). Chromium is believed to be a conservative contaminant, thus absorption and degradation can be neglected. The vertically-averaged model parameters are summarized as follows:

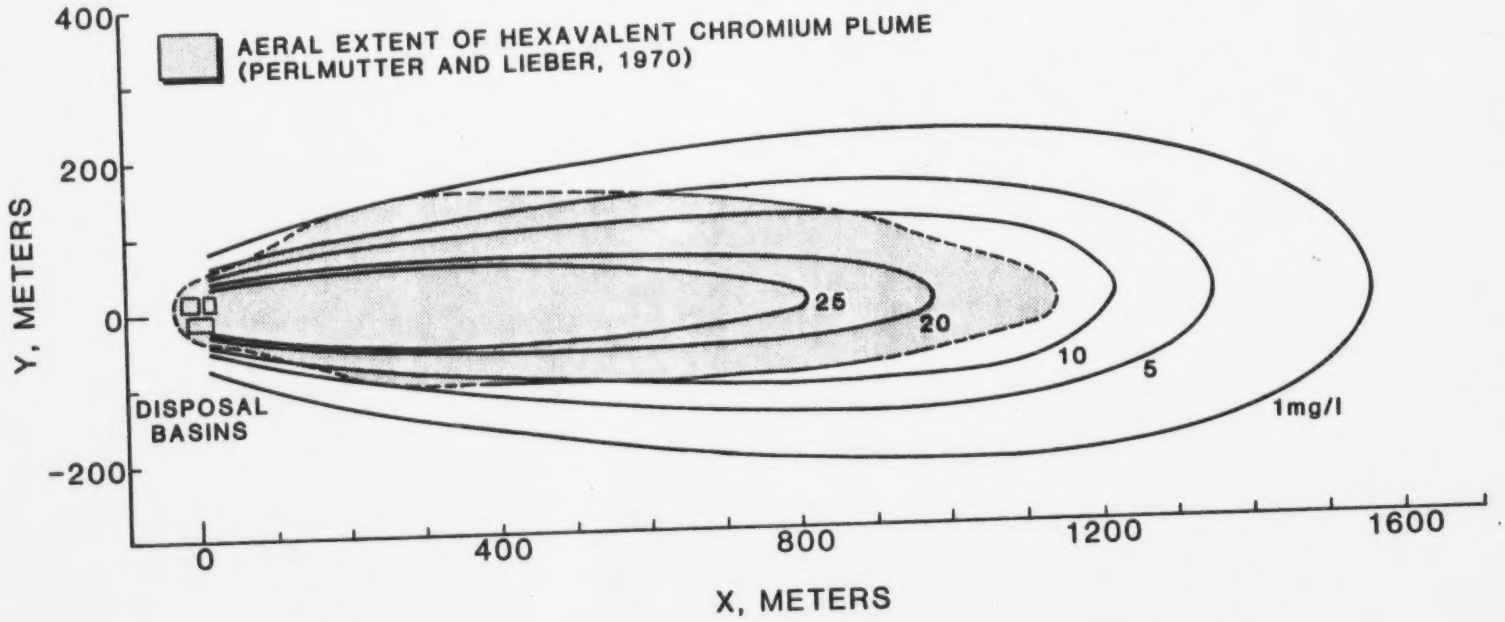


Figure 6. Results of hexavalent chromium plume simulation at 3280 days.

Aquifer porosity	0.35
Seepage velocity	0.366 m/dy
Retardation coefficient	1.0
x-Dispersion coefficient	7.79 m^2/dy
y-Dispersion coefficient	1.56 m^2/dy
Decay constant	0.0 1/dy

The contaminant source rate is assumed to be constant, and only one rate period is required. The mass rate can be converted to units of concentration times volume rate per unit depth as

$$\frac{23.6 \text{ kg}}{\text{dy}} \frac{10^6 \text{ mg}}{\text{kg}} \frac{\text{m}^3}{10^3 \text{ l}} \frac{1}{33.52 \text{ m}} = 704 \text{ (mg/l) } (\text{m}^3/\text{dy})/\text{m}$$

for approximately nine years or 3280 days.

The numerical results for the vertically-averaged solution are summarized in Figure 6. The shape and general extent of the predicted plume are in fair agreement with the observed extent of contamination considering the availability of field data and the assumptions which have been made in characterizing the problem as two-dimensional uniform flow with a continuous line source.

Superposition in time will be illustrated using data for the aquifer contaminated with chromium described above. Rather than a continuous source of contamination from the disposal ponds, as "accidental spill" of high strength waste will be simulated. The contaminant source will be assumed to be an "instantaneous line source" of strength 704 (mg/l) (m^3/m). The source rate schedule for the vertically-averaged model is constructed as follows:

Rate 1: 704 (mg/1) (m^3/dy)/m from 0 to 1 day

Rate 2: 0 (mg/1) (m^3/dy)/m from 1 to 365 days

Other model parameters are identical to those used in the previous example. The results of the simulation are summarized in Figure 7, which shows the center of mass of the plume moving down-gradient at the seepage velocity and spreading longitudinally and transversely by diffusion.

The results of the simulation using superposition in time were compared with the concentrations calculated using

$$C = \frac{C_o Q'}{4\pi\theta t (D_x D_y)^{0.5}} \text{EXP} \left(- \frac{(x - V^* t)^2}{4D_x t} - \frac{y^2}{4D_y t} - \lambda t\right) \quad (54)$$

which is the solution of Equation 14 for an instantaneous line source of strength $C_o Q'$ (m/L^3) (L^3/L). The values of concentration and errors in approximating the instantaneous source through superposition in time are presented in Table 2. Note that the finite duration of the source results is slightly higher concentrations up-gradient from the center of mass than concentrations down-gradient. For an instantaneous source the concentration distribution should be symmetrical about a y-z plane through the center of mass located as $x = Vt$. A better approximation can be obtained by injecting the same total mass of contaminant over a shorter period of time; but for purposes of illustrating superposition in time, the errors in the example problem are not significant.

The example problems presented above are intended to illustrate the application of the two-dimensional plume models developed in this report. These models are tools which can aid in the analysis of ground-water contamination problems. The user must select the best tool for the problem at

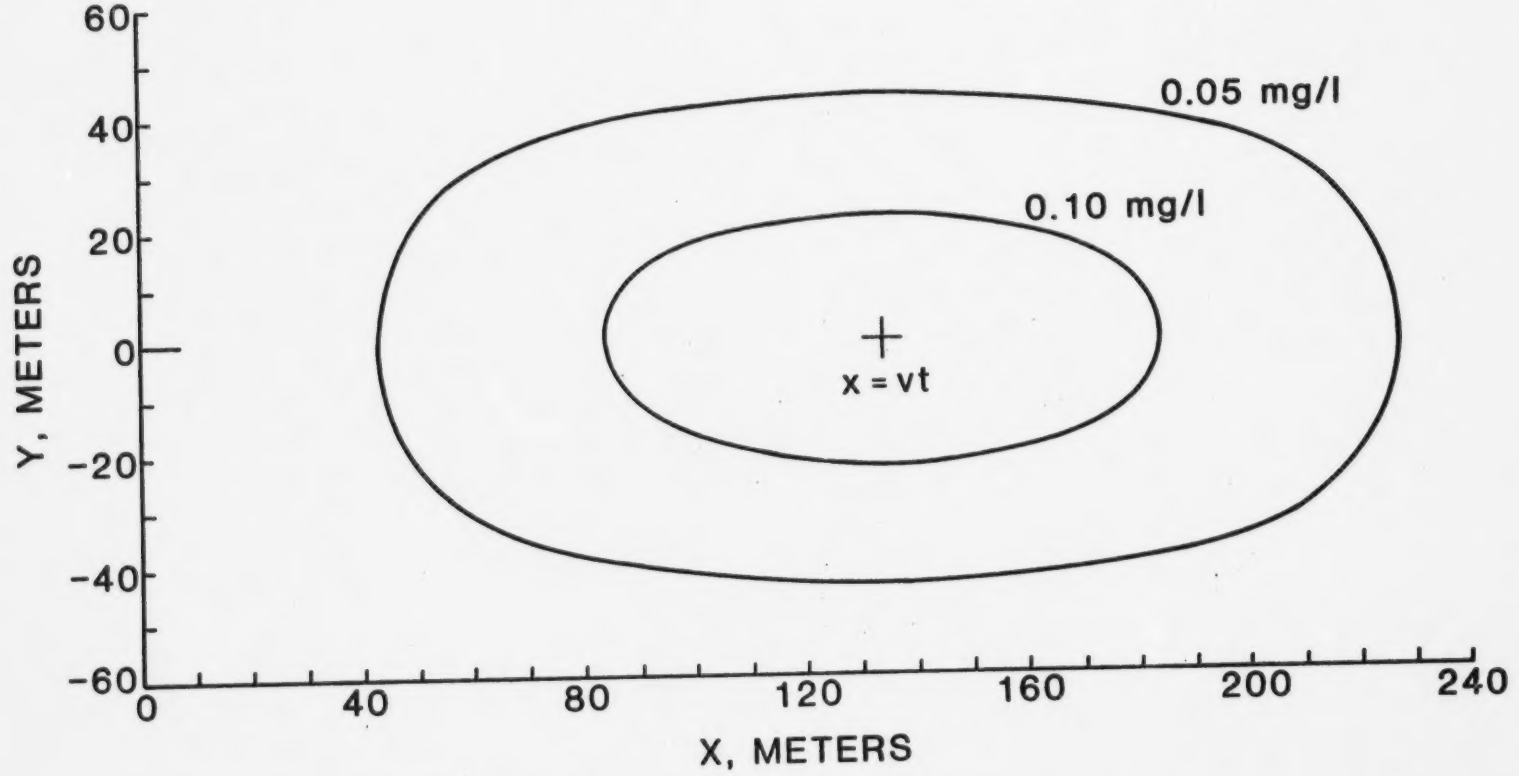


Figure 7. Results of hexavalent chromium spill simulation at 365 days.

hand, based on a sound understanding of the principles of ground-water hydrology, the physical problem, and the limitations of the mathematical model(s).

Perhaps the most difficult step in using any mathematical model is defining the problem to be solved. In addition to developing the physical boundaries of the problem domain, rock and fluid properties must also be quantified. Typical values of aquifer properties are listed in Table 3, but the user must accept the responsibility for developing the required model input data for the specific problem to be solved.

Table 2

COMPARISON OF CONCENTRATIONS CALCULATED USING SUPERPOSITION IN TIME AND AN ANALYTICAL SOLUTION FOR AN INSTANTANEOUS LINE SOURCE

MODEL PARAMETERS

Aquifer Porosity	0.35
Seepage Velocity	0.366 m/dy
Retardation Coefficient	1.0
x-Dispersion Coefficient	7.79 m^2/dy
y-Dispersion Coefficient	1.56 m^2/dy
First-Order Decay Constant	0.0 1/dy
Source Strength	704.0 (mg/1) (m^3/m)

Concentration at 365 days, mg/1

Superposition
(Equation 54)
% Error

y (meters)	x(meters) 73.59	103.59	133.59 (= Vt)	163.59	193.59
20.0	0.0919 (.0917) .22	0.1165 (.1162) .26	0.1259 (.1258) .08	0.1163 (.1162) .01	0.0916 (.0917) -.11
10.0	0.0878 (.0877) .11	0.1115 (.1112) .27	0.1206 (.1204) .17	0.1113 (.1112) .09	0.0876 (.0877) -.11
0.0	0.0771 (.0769) .26	0.0977 (.0975) .21	0.1057 (.1055) .19	0.0975 (.0975) .0	0.0768 (.0769) -.13

Table 3

Typical Values of Aquifer Properties

(after Yeh, 1981)

Parameter	Material		
	Clay	Silt	Sand
Bulk density, lb/ft^3	87.36 - 137.2	80.50 - 112.3	73.63 - 98.59
Effective porosity	0.03 - 0.05	0.05 - 0.10	0.10 - 0.30
Hydraulic Conductivity, $gal/day/ft^2$	0.01 - 0.1	1 - 10	100 - 100,000
Dispersivity, ft			
Longitudinal	0.1 - 1.0	1 - 10	10 - 100
Transverse	0.01 - 0.1	0.1 - 1.0	1.0 - 10
Vertical 0.01 - 0.1	0.1 - 1.0	1 - 10	

REFERENCES

Abramowitz, M. and I. A. Stegan, 1966, Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, National Bureau of Standards Applied Mathematics Series 55, U. S. Department of Commerce, 1046 pp.

Hantush, M. S., 1956, "Analysis of Data from Pumping Tests in Leaky Aquifers," Transitions, American Geophysical Union, Vol. 37, No. 6, pp. 702-714.

Hantush, M. S., 1964, "Hydraulics of Wells," Advances in Hydroscience, Vol. 1, pp. 281-432.

Hantush, M. S. and C. E. Jacob, 1955, "Non-Steady Radial Flow in an Infinite Leaky Aquifer," Transitions, American Geophysical Union, Vol. 26., No. 1, pp. 95-100.

Hunt, B., 1978, "Dispersive Sources in Uniform Ground-Water Flow," Journal of The Hydraulics Division, ASCE, Vol. 104, No. HY1, pp. 75-85.

Perlmutter, N. M. and M. Lieber, 1970, "Dispersal of Plating Wastes and Sewage Contaminants in Ground Water and Surface Water, South Farmingdale-Massapequa Area, Nassau County, New York," U. S. Geological Survey Water-Supply Paper 1879-G, pp. G1-G67.

Pinder, G. F., 1973, "A Galerkin Finite Element Simulation of Ground-water Contamination on Long Island," Water Resources Research, Vol. 9, No. 6, pp. 1657-1669.

Walton, W. C., 1962, "Selected Analytical Methods for Well and Aquifer Evaluation," Bulletin 49, Illinois State Water Survey, Urbana, Illinois, 81 pp.

Walton, W. C., 1970, Groundwater Resource Evaluation, Mc-Graw-Hill, New York, New York, 664 pp.

Wilson, J. L. and P. J. Miller, 1978, "Two-Dimensional Plume in Uniform Ground-Water Flow," Journal of the Hydraulics Division, ASCE, Vol. 104, No. HY4, pp. 503-514.

Wilson, J. L. and P. J. Miller, 1979, "Two-Dimensional Plume in Uniform Ground-Water Flow, Discussion," Journal of the Hydraulics Division, ASCE, Vol. 105, No. HY12, pp. 1567-1570.

Yeh, G. T., 1981, "AT123D: Analytical Transient One-, Two-, and Three-Dimensional Simulations of Waste Transport in the Aquifer System," Publication No. 1439, Environmental Sciences Division, Oak Ridge National Laboratory, Oak Ridge, Tennessee, 83 pp.

APPENDIX A

Example Problems

The two example problems presented in the following pages are discussed in Section III of this report. The first demonstrates the application of PLUME2D to a continuous source of contamination. The second example approximates an instantaneous source using the principle of superposition in time as discussed in Section I.

```
ENTER TITLE
?HEXAVALENT CHROMIUM PLUME


ENTER COORDINATE SYSTEM
   XY FOR VERTICALLY-AVERAGED SOLUTION
   XZ FOR HORIZONTALLY-AVERAGED SOLUTION
?XY


ENTER UNITS FOR LENGTH (2 CHARACTERS)
? M


ENTER UNITS FOR TIME (2 CHARACTERS)
?DY


ENTER UNITS FOR CONCENTRATION (6 CHARACTERS)
?MG/L


ENTER AQUIFER POROSITY
?Ø.35


ENTER SEEPAGE VELOCITY,  M/DY
?Ø.366


ENTER RETARDATION COEFFICIENT
?1.Ø


ENTER X DISPERSION COEFFICIENT, SQ  M/DY
?7.79


ENTER Y DISPERSION COEFFICIENT, SQ  M/DY
?1.56


ENTER DECAY CONSTANT, 1/DY
?Ø.Ø


SELECT TRANSIENT OR STEADY-STATE SOLUTION
   TR FOR TRANSIENT SOLUTION
   SS FOR STEADY-STATE SOLUTION
?TR
```

```
ENTER THE NUMBER OF SOURCES (MAXIMUM OF 10 )
?1


MASS RATES HAVE UNITS OF (MG/L  ) (CU  M/DY)
TIME HAS UNITS OF DY

ENTER X AND Y COORDINATES OF SOURCE 1 ( M)
?,?0.,0.


ENTER THE NUMBER RATES FOR SOURCE 1 (MAXIMUM 0F 10)
?1


SOURCE  1, RATE  1 STARTS AT     0.0 DY
ENTER MASS RATE AND ENDING TIME
?,?704.,3280.


ENTER XFIRST, XLAST, DELTAX  ( M)
?,?,?200.,1200.,200.


ENTER YFIRST, YLAST, DELTAY  ( M)
?,?,?200.,-200.,50.


ENTER TFIRST, TLAST, DELTAT  (DY)
?,?,?3280.,0.,0.
```

PLUME2D
VERSION 2.01
PAGE 1

HEXAVALENT CHROMIUM PLUME

SEEPAGE VELOCITY, (M/DY)	.3660
X DISPERSION COEFFICIENT (M**2/DY)	7.7900
Y DISPERSION COEFFICIENT (M**2/DY)	1.5600
POROSITY	.3500
RETARDATION COEFFICIENT	1.0000
FIRST ORDER DECAY CONSTANT (1/DY)	0.0000

SOURCE/RATE SCHEDULE (MG/L)(CU M/DY)

SOURCE NO	SOURCE X (M)	SOURCE Y (M)	RATE NO	MASS RATE	TIME (DY) START	TIME (DY) END
1	0.00	0.00	1	704.00	0.00	3280.00

OBSERVATION POINTS (M)

XFIRST =	200.00	XLAST =	1200.00	DELX =	200.0000
YFIRST =	200.00	YLAST =	-200.00	DELY =	50.0000

OBSERVATION TIMES (DY)

TFIRST =	3280.00	TLAST =	3280.00	DELT =	0.0000

```
 MENU OF EDIT COMMANDS

RETARDATION COEFFICIENT   RD      OBSERVATION POINTS   OB
POROSITY                  PO      X COORDINATES        XC
SEEPAGE VELOCITY          VX      Y COORDINATES        YC
X DISPERSION COEFFICIENT  DX      MENU OF COMMANDS     MU
Y DISPERSION COEFFICIENT  DY      LIST INPUT DATA      LI
DECAY CONSTANT            DE      RUN CALCULATIONS     RN
SOURCE RATE SCHEDULE      RT      DONE                 DN
NEW PROBLEM               NP      SATURATED THICKNESS  ST
CHANGE SOLUTION/SOURCES   CS      OBSERVATION TIMES    TC


 ENTER NEXT COMMAND
 ?RN
```

PLUME2D
VERSION 2.01
PAGE 2

HEXAVALENT CHROMIUM PLUME

CONCENTRATION DISTRIBUTION AT 3280.00 DY (MG/L)

* X(M) * / Y(M) *	200.00	400.00	600.00	800.00	1000.00	1200.00
200.00	.0372	.2773	.8210	1.4371	1.6352	1.1380
150.00	.4289	1.8560	3.6177	4.8444	4.7217	3.0238
100.00	4.0806	8.8387	11.3609	11.9818	10.2348	6.1201
50.00	24.5165	25.3968	23.5539	20.9946	16.4014	9.3721
0.00	51.8245	37.0664	30.2812	25.3930	19.2190	10.8087
-50.00	24.5165	25.3968	23.5539	20.9946	16.4014	9.3721
-100.00	4.0806	8.8387	11.3609	11.9818	10.2348	6.1201
-150.00	.4289	1.8560	3.6177	4.8444	4.7217	3.0238
-200.00	.0372	.2773	.8210	1.4371	1.6352	1.1380

ENTER NEXT COMMAND
?DN

STOP

```
ENTER TITLE
?ACCIDENTAL HEXAVALENT CHROMIUM SPILL


ENTER COORDINATE SYSTEM
   XY FOR VERTICALLY-AVERAGED SOLUTION
   XZ FOR HORIZONTALLY-AVERAGED SOLUTION
?XY


ENTER UNITS FOR LENGTH (2 CHARACTERS)
? M


ENTER UNITS FOR TIME (2 CHARACTERS)
?DY


ENTER UNITS FOR CONCENTRATION (6 CHARACTERS)
?MG/L


ENTER AQUIFER POROSITY
?Ø.35


ENTER SEEPAGE VELOCITY,  M/DY
?Ø.366


ENTER RETARDATION COEFFICIENT
?1.Ø


ENTER X DISPERSION COEFFICIENT, SQ  M/DY
?7.79


ENTER Y DISPERSION COEFFICIENT, SQ  M/DY
?1.56


ENTER DECAY CONSTANT, 1/DY
?Ø.Ø


SELECT TRANSIENT OR STEADY-STATE SOLUTION
   TR FOR TRANSIENT SOLUTION
   SS FOR STEADY-STATE SOLUTION
?TR
```

```
ENTER THE NUMBER OF SOURCES (MAXIMUM OF 10 )
?1


MASS RATES HAVE UNITS OF (MG/L  ) (CU  M/DY)
TIME HAS UNITS OF DY

ENTER X AND Y COORDINATES OF SOURCE 1 ( M)
?,?0.,0.


ENTER THE NUMBER RATES FOR SOURCE 1 (MAXIMUM 0F 10)
?2


SOURCE  1, RATE  1 STARTS AT     0.0 DY
ENTER MASS RATE AND ENDING TIME
?,?704.,1.


SOURCE  1, RATE  2 STARTS AT     1.0 DY
ENTER MASS RATE AND ENDING TIME
?,?0.,365.


ENTER XFIRST, XLAST, DELTAX  ( M)
?,?,?73.59,193.59,30.


ENTER YFIRST, YLAST, DELTAY  ( M)
?,?,?20.,0.,10.


ENTER TFIRST, TLAST, DELTAT  (DY)
?,?,?365.,0.,0.
```

PLUME2D
VERSION 2.01
PAGE 1

ACCIDENTAL HEXAVALENT CHROMIUM SPILL

SEEPAGE VELOCITY, (M/DY)	.3660
X DISPERSION COEFFICIENT (M**2/DY)	7.7900
Y DISPERSION COEFFICIENT (M**2/DY)	1.5600
POROSITY	.3500
RETARDATION COEFFICIENT	1.0000
FIRST ORDER DECAY CONSTANT (1/DY)	0.0000

SOURCE/RATE SCHEDULE (MG/L)(CU M/DY)

SOURCE		RATE	MASS	TIME (DY)	
NO	X (M)	Y (M)	NO	RATE	START	END
1	0.00	0.00				
			1	704.00	0.00	1.00
			2	0.00	1.00	365.00

OBSERVATION POINTS (M)

XFIRST =	73.59	XLAST =	193.59	DELX =	30.0000
YFIRST =	20.00	YLAST =	0.00	DELY =	10.0000

OBSERVATION TIMES (DY)

TFIRST =	365.00	TLAST =	365.00	DELT =	0.0000

```
 MENU OF EDIT COMMANDS

RETARDATION COEFFICIENT  RD      OBSERVATION POINTS  OB
POROSITY                 PO      X COORDINATES       XC
SEEPAGE VELOCITY         VX      Y COORDINATES       YC
X DISPERSION COEFFICIENT DX      MENU OF COMMANDS    MU
Y DISPERSION COEFFICIENT DY      LIST INPUT DATA     LI
DECAY CONSTANT           DE      RUN CALCULATIONS    RN
SOURCE RATE SCHEDULE     RT      DONE                DN
NEW PROBLEM              NP      SATURATED THICKNESS ST
CHANGE SOLUTION/SOURCES  CS      OBSERVATION TIMES   TC


 ENTER NEXT COMMAND
 ?RN
```

ACCIDENTAL HEXAVALENT CHROMIUM SPILL

CONCENTRATION DISTRIBUTION AT 365.00 DY (MG/L)

Y(M) \ X(M)	73.59	103.59	133.59	163.59	193.59
20.00	.0771	.0977	.1056	.0975	.0768
10.00	.0879	.1115	.1204	.1113	.0876
0.00	.0919	.1165	.1260	.1163	.0916

ENTER NEXT COMMAND
?XC

ENTER XFIRST, XLAST, DELTAX (M)
?,?,?103.59,163.59,15.

ENTER NEXT COMMAND
?RN

PLUME2D
VERSION 2.01
PAGE 3

ACCIDENTAL HEXAVALENT CHROMIUM SPILL

CONCENTRATION DISTRIBUTION AT 365.00 DY (MG/L)

Y(M) \ X(M)	103.59	118.59	133.59	148.59	163.59
20.00	.0977	.1036	.1056	.1035	.0975
10.00	.1115	.1183	.1204	.1181	.1113
0.00	.1165	.1236	.1260	.1234	.1163

ENTER NEXT COMMAND
?DN

STOP

APPENDIX B

Description of Program PLUME2D

Program PLUME2D has been written in an unextended Fortran computer code in an effort to make the program transportable between computer systems. The computer code consists of a main program and several function subroutines which are required to evaluate the Hantush well function. The program has been documented "internally" through the liberal use of comment statements.

The main program has been divided into three sections. A listing of the computer code is presented in Appendix D. Section I provides for the "Basic Input Data" as described in Section II of this report. The numerical evaluation of concentration at specified grid coordinates is accomplished in Section II of the main program which calls subroutine SOL2D, the code for the analytical solution of the governing differential equations. Section III provides for problem redefinition and control of execution under the "Edit" mode discussed in the body of this report.

Ten function subroutines are used to evaluate the Hantush well function using the numerical methods described in Appendix C. Listings of the computer codes are presented in Appendix E. FUNCTION W(U,B) evaluates the Hantush well function for $B < 20$. For $B > 20$, the term $EXP(Pe_x/2)\ W(U,B)$ in Equation 45 is evaluated using FUNCTION WELPRD(U,B,PEX). This procedure is used to avoid taking the direct product of very large numbers, $EXP(Pe_x/2)$, and very small numbers W(U,B), for large values of B.

FUNCTION GAUSS is a 24-point Gauss-Legendre quadrature numerical integration scheme which is used to evaluate the Hantush well function using either Equation C-6 or Equation C-7. FUNCTION FUNCTN evaluates the integrand of Equations C-6 and C-7.

The six remaining function subroutines are used to evaluate mathematical functions using rational approximations or polynomial approximations. They are:

FUNCTION BIO(Z) FUNCTION BIOLOG(Z)	Modified Bessel function of the first kind of order zero and the natural logarithm of the function.
FUNCTION BKO(Z) FUNCTION BKOLOG(Z)	Modified Bessel function of the second kind of order zero and the natural logarithm of the function.
FUNCTION E1LOG(Z)	Natural logrithm of the exponential integral.
FUNCTION ERFC(Z)	Complimentary error function.

These six function subroutines are used to support FUNCTION W(U,B) and/or FUNCTION WELPRD (U,B,PEX). If system subroutines are available for these functions they may be substituted for the function subroutines provided with Program PLUME2D.

APPENDIX C

Numerical Evaluation of the Hantush Well Function

The Hantush well function can be defined as

$$W(U,B) = \int_u^\infty \frac{1}{\xi} \text{EXP}\left(- \xi - \frac{B^2}{4\xi}\right) d\xi \qquad \text{(C-1)}$$

or the reciprocal relation

$$W(U,B) = 2K_o(B) - \int_{B^2/4U}^\infty \frac{1}{\xi} \text{EXP}\left(- \xi - \frac{B^2}{4\xi}\right) d\xi \qquad \text{(C-2)}$$

where ξ is a dummy integration variable (Hantush, 1964). Using the identity

$$\int_a^\infty f(\xi)\, d\xi = \int_o^\infty f(\xi)\, d\xi - \int_o^a f(\xi)\, d\xi \qquad \text{(C-3)}$$

Equation C-1 can be rewritten as

$$W(U,B) = \int_o^\infty \frac{1}{\xi} \text{EXP}\left(- \xi - \frac{B^2}{4\xi}\right) d\xi - \int_o^u \frac{1}{\xi} \left(- \xi - \frac{B^2}{4\xi}\right) d\xi \qquad \text{(C-4)}$$

Now

$$\int_o^\infty \frac{1}{\xi} \text{EXP}\left(- \xi - \frac{B^2}{4\xi}\right) d\xi = 2K_o(B) \qquad \text{(C-5)}$$

where K_o is the modified Bessel function of the second kind of order zero. Substituting Equation C-5 into Equation C-4, the well function becomes

$$W(U,B) = 2K_o(B) - \int_o^u \frac{1}{\xi} EXP(- \xi - \frac{B^2}{4\xi}) \, d\xi \qquad (C-6)$$

The reciprocal relation, Equation C-2, can also be written in terms of finite limits. Using the relationship given by Equations C-4 and C-5, the reciprocal relation can be expressed as

$$W(U,B) = \int_o^{B^2/4U} \frac{1}{\xi} EXP(- \xi - \frac{B^2}{4\xi}) \, d\xi \qquad (C-7)$$

For $0 < B \leq 20$, values of $W(U,B)$ for $0 < U < B/2$ are obtained from Equation C-6 by first evaluating the value of the integrand using a 24-point Gauss-Legendre numerical integration scheme. For $B/2 \leq U < \infty$, the reciprocal relation, Equation C-7, is evaluated using the same numerical integration scheme.

For $0 < B \leq 0.1$, values of $W(U,B)$ are obtained from the series expansions presented by Hantush and Jacob (1955). For $U \leq 1$

$$W(U,B) = 2K_o(B) - I_o(B)E_1(\frac{B^2}{4U})$$

$$+ EXP(- \frac{B^2}{4U}) \left[0.57721566 + \ln(U) + E_1(U) + \frac{U}{4}\frac{B^2}{4}(1 - \frac{U}{9})\right] \qquad (C-8)$$

and for $U > 1$

$$W(U,B) = I_o(B)E_1(U) - EXP(-U)\frac{B^2}{4}\left[(\frac{1}{U} - \frac{1}{36U^2}) + \frac{B^4}{16}(\frac{1}{4U} - \frac{1}{4U^2})\right] \qquad (C-9)$$

where I_o is the modified Bessel function of the first kind of order zero, E_1 is the exponential integral, and 0.57721566 is Euler's constant.

For B > 20, the third order approximation for W(U,B) presented by Wilson and Miller (1979) is used to evaluate the well function. The approximation is

$$W(U,B) = \left(\frac{\pi}{2B}\right)^{1/2} \text{EXP}(-B) \left[\left(1 - \frac{1}{8B}\right) \text{ERFC}\ (-\beta) + \frac{\beta}{4B\pi^{1/2}} \text{EXP}(-\beta^2)\right] \tag{C-10}$$

where

$$\beta = \frac{B-2U}{(4U)^{1/2}}$$

and ERFC is the complimentary error function.

Now, for large positive values of β,

$$W(U,B) \simeq 2 \left(\frac{\pi}{2B}\right)^{1/2} \text{EXP}(-B) \left(1 - \frac{1}{8B}\right) \tag{C-11}$$

and an asympotic expansion for $K_o(B)$ can be written as

$$K_o(B) = \left(\frac{\pi}{2B}\right)^{1/2} \text{EXP}(-B) \left(1 - \frac{1}{8B} + \frac{9}{2\ (8B)^2} + \ldots\right) \tag{C-12}$$

Thus for B > 20 and β > 7.5 the well function is approximated as

$$W(U,B) = 2K_o(B) \tag{C-13}$$

Note that this approximation is equivalent to the relationship

$$W(0,B) = 2K_o(B) \tag{C-14}$$

Evaluations of the Hantush well function using the methods described in the previous paragraphs have been checked for both accuracy and continuity of the function between the various approximations. The Gauss-Legendre quadrature scheme was checked using up to 48 quadrature points. A maximum of 24 quadrature points yielded results accurate to four significant figures in the mantissa over the entire range of arguments which require numerical integration. The other approximations for W(U,B) are also accurate to four significant figures in the mantissa.

APPENDIX D

Listing of Program PLUME2D

```
C     PLUME2D                                                                  PL2D001
C     VERSION 2.02                                                             PL2D002
C     TWO-DIMENSIONAL PLUMES IN UNIFORM GROUND-WATER FLOW                      PL2D003
C        JAN WAGNER                                                            PL2D004
C        SCHOOL OF CHEMICAL ENGINEERING                                        PL2D005
C        OKLAHOMA STATE UNIVERSITY                                             PL2D006
C        STILLWATER, OK  74078                                                 PL2D007
C        PHONE (405) 624-5280                                                  PL2D008
C        JULY, 1981                                                            PL2D009
C                                                                              PL2D010
C     REVISIONS:    2.00       APR 84                                          PL2D011
C                   2.01    24 NOV 84                                          PL2D012
C                   2.02     9 DEC 84                                          PL2D013
C                   2.02A    3 MAY 85                                          PL2D014
C                                                                              PL2D015
      DIMENSION TITLE(30),IC(20),XS(10),YS(10),D(3),LBL(2,6),                  PL2D016
     1 NR(10),IS(4),NP(2),DEL(2),XL(2),XF(2),CON(7),COL(7)                     PL2D017
      REAL LAMBDA                                                              PL2D018
      INTEGER TITLE                                                            PL2D019
      COMMON/IO/NI,NO                                                          PL2D020
      COMMON/RATE/Q(10,12),T(10,12),MT                                         PL2D021
      COMMON/PHYPRO/ALPHA,BETA,DX,LAMBDA,PE,RD,V                               PL2D022
      DATA IC/'DE','VX','RD','DX','DY','DZ','PO','OB','XC','YC','ZC',         PL2D023
     1'RT','NP','RN','DN','LI','MU','ST','CS','TC'/                           PL2D024
      DATA KPRO1/'XY'/, KPRO2/'XZ'/, KHARY/'Y'/, KHARZ/'Z'/                    PL2D025
      DATA NPAGE/1/                                                            PL2D026
      DATA KSOL1,KSOL2/'TR','SS'/                                              PL2D027
      DATA IS/'R','M','A','D'/                                                 PL2D028
      DATA IY/'Y'/                                                             PL2D029
      DATA LBL/'  ','(C','  ','ON','  ','TI','  ','NU','  ','ED',              PL2D030
     1 '  ','.') '/                                                            PL2D031
C                                                                              PL2D032
C     READ DEVICE: NI       WRITE DEVICE: NO                                   PL2D033
      NI=5                                                                     PL2D034
      NO=6                                                                     PL2D035
C                                                                              PL2D036
C     MAXIMUM NUMBER OF PRINTED COLUMNS PER PAGE IS SET TO MAXCOL              PL2D037
C        DIMENSION COL(MAXCOL),CON(MAXCOL)                                     PL2D038
      MAXCOL = 7                                                               PL2D039
C                                                                              PL2D040
C     MAXIMUM NUMBER OF PRINTED ROWS PER PAGE IS SET TO MAXROW                 PL2D041
      MAXROW = 40                                                              PL2D042
C                                                                              PL2D043
C     MAXIMUM NUMBER OF SOURCES IS SET TO MAXSOR                               PL2D044
C        DIMENSION XS(MAXSOR),YS(MAXSOR),NR(MAXSOR)                            PL2D045
      MAXSOR = 10                                                              PL2D046
C                                                                              PL2D047
C     MAXIMUM NUMBER OF SOURCE RATES FOR SUPERPOSITION IN TIME                 PL2D048
C     IS SET TO MAXRT                                                          PL2D049
C        COMMON/RATE/ Q(MAXSOR,MAXRT+2),T(MAXSOR,MAXRT+2)                     PL2D050
      MAXRT = 10                                                               PL2D051
C                                                                              PL2D052
C     MAXIMUM NUMBER OF IMAGE WELLS FOR SUPERPOSITION IN SPACE                 PL2D053
C     IS SET TO MAXIMG                                                         PL2D054
      MAXIMG = 20                                                              PL2D055
C                                                                              PL2D056
C                                                                              PL2D057
C     INITIALIZE PROGRAM FLOW CONTROL VARIABLES                                PL2D058
    1 IEDIT = 1                                                                PL2D059
      KNTL = 1                                                                 PL2D060
C                                                                              PL2D061
C                                                                              PL2D062
C **** SECTION I -- BASIC INPUT DATA                                           PL2D063
C                                                                              PL2D064
C     READ TITLE                                                               PL2D065
      WRITE(NO,3)                                                              PL2D066
    3 FORMAT(1H1,2X,'ENTER TITLE',/'   ?')                                     PL2D067
      READ(NI,5) (TITLE(I), I=1,30)                                            PL2D068
    5 FORMAT(30A2)                                                             PL2D069
C                                                                              PL2D070
```

```
C     SELECT VERTICALLY OR HORIZONTALLLY AVERAGED SOLUTION                    PL2D071
      KFLOW = 3                                                               PL2D072
      WRITE(ND,7)                                                             PL2D073
    7 FORMAT(3X,'ENTER COORDINATE SYSTEM',/,                                  PL2D074
     16X,'XY FOR VERTICALLY-AVERAGED SOLUTION',/,                             PL2D075
     26X,'XZ FOR HORIZONTALLY-AVERAGED SOLUTION',/,'   ?')                    PL2D076
    8 READ(NI,9) KFLG                                                         PL2D077
    9 FORMAT(A2)                                                              PL2D078
      IF(KFLG.EQ.KPRO1) KFLOW=1                                               PL2D079
      IF(KFLG.EQ.KPRO2) KFLOW=2                                               PL2D080
      GO TO (12,12,10), KFLOW                                                 PL2D081
   10 WRITE(ND,11)                                                            PL2D082
   11 FORMAT(3X,'ERROR IN PROBLEM SELECTION -- REENTER',/,'   ?')             PL2D083
      GO TO 8                                                                 PL2D084
   12 CONTINUE                                                                PL2D085
      KHAR = KHARY                                                            PL2D086
      IF(KFLOW.EQ.2) KHAR=KHARZ                                               PL2D087
C                                                                             PL2D088
C     DEFINE UNITS                                                            PL2D089
      WRITE(ND,15)                                                            PL2D090
   15 FORMAT(3X,'ENTER UNITS FOR LENGTH (2 CHARACTERS)',/,'   ?')             PL2D091
      READ(NI,25) IL                                                          PL2D092
   25 FORMAT(A2)                                                              PL2D093
      WRITE(ND,35)                                                            PL2D094
   35 FORMAT(3X,'ENTER UNITS FOR TIME (2 CHARACTERS)',/,'   ?')               PL2D095
      READ(NI,25) IT                                                          PL2D096
      WRITE(ND,45)                                                            PL2D097
   45 FORMAT(3X,'ENTER UNITS FOR CONCENTRATION (6 CHARACTERS)',/,'   ?')      PL2D098
      READ(NI,26) IM1,IM2,IM3                                                 PL2D099
   26 FORMAT(3A2)                                                             PL2D100
C                                                                             PL2D101
C     ENTER DATA FOR FIRST PROBLEM                                            PL2D102
C                                                                             PL2D103
C     SATURATED THICKNESS                                                     PL2D104
      IF(KFLOW.EQ.1)GO TO 38                                                  PL2D105
   30 IMAGE = MAXIMG/2                                                        PL2D106
   44 WRITE(ND,46) IL                                                         PL2D107
   46 FORMAT(3X,'ENTER SATURATED THICKNESS (0 FOR INFINITE THICKNESS), '      PL2D108
     1A2,/,'   ?')                                                           PL2D109
   36 READ(NI,37,ERR=44) ST                                                   PL2D110
   37 FORMAT(F10.0)                                                           PL2D111
      IF(ST.GT.0.0) GO TO 39                                                  PL2D112
   38 IMAGE = 1                                                               PL2D113
      ST = 1.0E32                                                             PL2D114
   39 CONTINUE                                                                PL2D115
      GO TO (50,400),IEDIT                                                    PL2D116
C                                                                             PL2D117
C     POROSITY                                                                PL2D118
   50 WRITE(ND,55)                                                            PL2D119
   55 FORMAT(3X,'ENTER AQUIFER POROSITY',/,'   ?')                            PL2D120
      READ(NI,56,ERR=50) P                                                    PL2D121
   56 FORMAT(F10.0)                                                           PL2D122
   57 IF(P.GT.0.0.AND.P.LT.1.0) GO TO 59                                      PL2D123
   54 WRITE(ND,58)                                                            PL2D124
   58 FORMAT(3X,'POROSITY MUST BE GREATER THAN ZERO',                         PL2D125
     1' AND LESS THAN ONE -- REENTER',/,'   ?')                               PL2D126
      READ(NI,56,ERR=54) P                                                    PL2D127
      GO TO 57                                                                PL2D128
   59 GO TO (60,400),IEDIT                                                    PL2D129
C                                                                             PL2D130
C     SEEPAGE VELOCITY                                                        PL2D131
   60 WRITE(ND,65) IL,IT                                                      PL2D132
   65 FORMAT(3X,'ENTER SEEPAGE VELOCITY, 'A2,'/',A2,/,'   ?')                 PL2D133
      READ(NI,56,ERR=60) V                                                    PL2D134
   66 IF(V.GT.0.0) GO TO 69                                                   PL2D135
   64 WRITE(ND,67)                                                            PL2D136
   67 FORMAT(3X,'SEEPAGE VELOCITY MUST BE GREATER THAN ZERO',                 PL2D137
     1' -- REENTER',/,'   ?')                                                 PL2D138
      READ(NI,56,ERR=64) V                                                    PL2D139
      GO TO 66                                                                PL2D140
```

```
   69 GO TO (70,400),IEDIT                                                  PL2D141
C                                                                           PL2D142
C     RETARDATION COEFFICIENT                                               PL2D143
   70 WRITE(NO,75)                                                          PL2D144
   75 FORMAT(3X,'ENTER RETARDATION COEFFICIENT',/,'   ?')                   PL2D145
      READ(NI,56,ERR=70) RD                                                PL2D146
   76 IF(RD.GE.1.0) GO TO 79                                               PL2D147
   74 WRITE(NO,77)                                                          PL2D148
   77 FORMAT(3X,'RETARDATION COEFFICIENT MUST BE GREATER THAN OR',          PL2D149
     1' EQUAL TO ONE',/,' -- REENTER',/,'   ?')                             PL2D150
      READ(NI,56,ERR=74) RD                                                PL2D151
      GO TO 76                                                              PL2D152
   79 GO TO (80,400),IEDIT                                                  PL2D153
C                                                                           PL2D154
C     X DISPERSION COEFFICIENT                                              PL2D155
   80 WRITE(NO,81) IL,IT                                                    PL2D156
   81 FORMAT(3X,'ENTER X DISPERSION COEFFICIENT, SQ ',A2,                   PL2D157
     1'/',A2,/,'   ?')                                                     PL2D158
   82 READ(NI,56,ERR=80) DX                                                 PL2D159
      IF(DX.GT.0.0) GO TO 85                                                PL2D160
      WRITE(NO,83)                                                          PL2D161
   83 FORMAT(3X,'X DISPERSION COEFFICIENT MUST BE GREATER THAN ZERO',       PL2D162
     1' -- REENTER',/,'   ?')                                               PL2D163
      GO TO 82                                                              PL2D164
   85 GO TO (86,400),IEDIT                                                  PL2D165
C                                                                           PL2D166
C     Y OR Z DISPERSION COEFFICIENT                                         PL2D167
   86 WRITE(NO,87) KHAR,IL,IT                                               PL2D168
   87 FORMAT(3X,'ENTER ',A1,' DISPERSION COEFFICIENT, SQ ',A2,               PL2D169
     1'/',A2,/,'   ?')                                                     PL2D170
   88 READ(NI,56,ERR=86) DY                                                 PL2D171
      IF(DY.GT.0.0) GO TO 90                                                PL2D172
      WRITE(NO,89) KHAR                                                     PL2D173
   89 FORMAT(3X,A1,' DISPERSION COEFFICIENT MUST BE GREATER THAN ZERO',     PL2D174
     1' -- REENTER',/,'   ?')                                               PL2D175
      GO TO 88                                                              PL2D176
   90 GO TO (91,400),IEDIT                                                  PL2D177
C                                                                           PL2D178
C     FIRST-ORDER DECAY CONSTANT                                            PL2D179
   91 WRITE(NO,95) IT                                                       PL2D180
   95 FORMAT(3X,'ENTER DECAY CONSTANT, 1/',A2,/,'   ?')                     PL2D181
      READ(NI,56,ERR=91) DECAY                                              PL2D182
      GO TO (1320,400),IEDIT                                                PL2D183
C                                                                           PL2D184
C     SOURCE RATE SCHEDULE                                                  PL2D185
C                                                                           PL2D186
C                                                                           PL2D187
C     DEFINE LOCATIONS AND RATES OF SOURCES                                 PL2D188
C     INITIALIZE SOURCE/RATE ARRAYS                                         PL2D189
 1320 MAXRT2 = MAXRT + 2                                                    PL2D190
      DO 1330  I=1,MAXSOR                                                   PL2D191
        DO 1330  J=1,MAXRT2                                                 PL2D192
           Q(I,J) = 0.0                                                     PL2D193
           T(I,J) = 0.0                                                     PL2D194
 1330 CONTINUE                                                              PL2D195
      JFLOW = 3                                                             PL2D196
 1340 WRITE(NO,1345)                                                        PL2D197
 1345 FORMAT(3X,'SELECT TRANSIENT OR STEADY-STATE SOLUTION',/,              PL2D198
     16X,'TR FOR TRANSIENT SOLUTION',/,                                     PL2D199
     26X,'SS FOR STEADY-STATE SOLUTION',/,'   ?')                          PL2D200
 1350 READ(NI,25) KSOL                                                      PL2D201
      IF(KSOL.EQ.KSOL1) JFLOW=1                                             PL2D202
      IF(KSOL.EQ.KSOL2) JFLOW=2                                             PL2D203
      GO TO (1370,1370,1360), JFLOW                                         PL2D204
 1360 WRITE(NO,1365)                                                        PL2D205
 1365 FORMAT(3X,'ERROR IN SELECTION -- REENTER'/,'   ?')                    PL2D206
      GO TO 1350                                                            PL2D207
C                                                                           PL2D208
 1370 WRITE(NO,1375) MAXSOR                                                 PL2D209
 1375 FORMAT(3X,'ENTER THE NUMBER OF SOURCES (MAXIMUM OF',I3,' )',/,        PL2D210
```

```
     1'   ?')                                                              PL2D211
1380 READ(NI,1385,ERR=1370) FDUM                                          PL2D212
1385 FORMAT(F10.0)                                                        PL2D213
     NS=FDUM                                                              PL2D214
     IF(NS.GT.0.AND.NS.LE.MAXSOR) GO TO 1400                              PL2D215
     WRITE(NO,1395) MAXSOR                                                PL2D216
1395 FORMAT(3X,'NUMBER OF SOURCES MUST BE GREATER THAN ZERO '             PL2D217
    1'AND LESS THAN',I3,' -- REENTER',/,'   ?')                           PL2D218
     GO TO 1380                                                           PL2D219
1400 WRITE (NO,1405) IM1,IM2,IM3,IL,IT,IL,IT                              PL2D220
1405 FORMAT(3X,'MASS RATES HAVE UNITS OF (',3A2,') (CU ',A2,'/',A2,       PL2D221
    1 ')/',A2,/,3X,'TIME HAS UNITS OF ',A2,/)                             PL2D222
     DO 1540  I=1,NS                                                      PL2D223
     IF(KFLOW.EQ.2) GO TO 1414                                            PL2D224
1406 WRITE(NO,1410) I,IL                                                  PL2D225
1410 FORMAT(3X,'ENTER X AND Y COORDINATES OF SOURCE',I2,                  PL2D226
    1' (',A2,')',/,'   ?,?')                                             PL2D227
     READ(NI,1425,ERR=1406) XS(I),YS(I)                                   PL2D228
     GO TO 1440                                                           PL2D229
1414 WRITE(NO,1415) I,IL                                                  PL2D230
1415 FORMAT(3X,'ENTER X AND Z COORDINATES OF SOURCE',I2,                  PL2D231
    1' (',A2,')',/,'   ?,?')                                             PL2D232
     READ(NI,1425,ERR=1414) XS(I),YS(I)                                   PL2D233
1425 FORMAT(2F10.0)                                                       PL2D234
1430 IF(YS(I).GE.0.0.AND.YS(I).LE.ST) GO TO 1440                          PL2D235
1434 WRITE(NO,1435) ST,IL                                                 PL2D236
1435 FORMAT(3X,'Z-COORDINATE MUST BE GREATER THAN OR EQUAL TO ZERO',      PL2D237
    1' AND',/,3X,'LESS THAN OR EQUAL TO SATURATED THICKNESS (',           PL2D238
    2F10.4,A3,')',/,3X,' -- REENTER',/,'   ?')                            PL2D239
     READ(NI,37,ERR=1434) YS(I)                                           PL2D240
     GO TO 1430                                                           PL2D241
1440 IF(JFLOW.EQ.2) GO TO 1530                                            PL2D242
     Q(I,1) = 0.0                                                         PL2D243
     T(I,1) = 0.0                                                         PL2D244
1450 WRITE(NO,1455) I,MAXRT                                               PL2D245
1455 FORMAT(3X,'ENTER THE NUMBER RATES FOR SOURCE',I2,                    PL2D246
    1' (MAXIMUM OF',I3,')',/,'   ?')                                      PL2D247
1460 READ(NI,1465,ERR=1450) FDUM                                          PL2D248
1465 FORMAT(F10.0)                                                        PL2D249
     NR(I)=FDUM                                                           PL2D250
     IF(NR(I).GT.0.AND.NR(I).LE.MAXRT) GO TO 1480                         PL2D251
     WRITE(NO,1475) MAXRT                                                 PL2D252
1475 FORMAT(3X,'NUMBER OF RATES MUST BE GREATER THAN ZERO AND ',          PL2D253
    1'LESS THAN',I3,'  -- REENTER',/,'   ?')                              PL2D254
     GO TO 1460                                                           PL2D255
1480 CONTINUE                                                             PL2D256
     NRT = NR(I)                                                          PL2D257
     DO 1520  J=1,NRT                                                     PL2D258
        M = J + 1                                                         PL2D259
1484    WRITE(NO,1485) I,J,T(I,M-1),IT                                    PL2D260
1485    FORMAT(3X,'SOURCE ',I2,', RATE ',I2,' STARTS AT',F8.1,A3,/,       PL2D261
    1   3X,'ENTER MASS RATE AND ENDING TIME ',/,'   ?,?')                 PL2D262
        READ(NI,1495,ERR=1484) Q(I,M),T(I,M)                              PL2D263
1495    FORMAT(2F10.0)                                                    PL2D264
1500    IF(T(I,M).GT.T(I,M-1)) GO TO 1510                                 PL2D265
1504    WRITE(NO,1505)                                                    PL2D266
1505    FORMAT(3X,'ENDING TIME MUST BE GREATER THAN STARTING TIME '       PL2D267
    1   ' -- REENTER',/,'   ?')                                           PL2D268
        READ(NI,37,ERR=1504) T(I,M)                                       PL2D269
        GO TO 1500                                                        PL2D270
1510    CONTINUE                                                          PL2D271
1520 CONTINUE                                                             PL2D272
     GO TO 1540                                                           PL2D273
1530 WRITE(NO,1535) I                                                     PL2D274
1535 FORMAT(3X,'ENTER STEADY-STATE MASS RATE',I2,/,'   ?')                PL2D275
     READ(NI,37,ERR=1530) Q(I,1)                                          PL2D276
     NR(I) = 0                                                            PL2D277
1540 CONTINUE                                                             PL2D278
     IF(IEDIT.EQ.2.AND.JFLOW.EQ.1.AND.TF.LE.1.0E-06) GO TO 720            PL2D279
 123 GO TO (124,400),IEDIT                                                PL2D280
```

```
C                                                                       PL2D281
C     COORDINATES OF THE OBSERVATION POINTS                             PL2D282
  124 WRITE(NO,125) IL                                                  PL2D283
  125 FORMAT(3X,'ENTER XFIRST, XLAST, DELTAX  (',A2,')' ,/,'   ?,?,?')  PL2D284
      READ(NI,126,ERR=124) XF(1),XL(1),DEL(1)                           PL2D285
  126 FORMAT(3F10.0)                                                    PL2D286
      DEL(1) = ABS(DEL(1))                                              PL2D287
      IF(DEL(1).LE.1.0E-06) XL(1)=XF(1)                                 PL2D288
      IF(KNTL.LE.0) GO TO 400                                           PL2D289
  133 IF(KFLOW.EQ.2) GO TO 136                                          PL2D290
  134 WRITE(NO,135) IL                                                  PL2D291
  135 FORMAT(3X,'ENTER YFIRST, YLAST, DELTAY  (',A2,')',/,'   ?,?,?')   PL2D292
      READ(NI,126,ERR=134) XF(2),XL(2),DEL(2)                           PL2D293
      DEL(2) = ABS(DEL(2))                                              PL2D294
      IF(DEL(2).LE.1.0E-06) XL(2)=XF(2)                                 PL2D295
      GO TO 145                                                         PL2D296
  136 WRITE(NO,137) IL                                                  PL2D297
  137 FORMAT(3X,'ENTER ZFIRST, ZLAST, DELTAZ  (',A2,')',/,'   ?,?,?')   PL2D298
      READ(NI,126,ERR=136) XF(2),XL(2),DEL(2)                           PL2D299
      DEL(2) = ABS(DEL(2))                                              PL2D300
  138 IF(XF(2).GE.0.0.AND.DEL(2).LE.1.0E-06) GO TO 144                  PL2D301
      IF(XF(2).GE.0.0.AND.XF(2).LE.ST) GO TO 142                        PL2D302
      WRITE(NO,139)                                                     PL2D303
  139 FORMAT(3X,'ZFIRST MUST BE GREATER THAN OR EQUAL TO ZERO')         PL2D304
      IF(IMAGE.GT.1) WRITE(NO,140) ST,IL                                PL2D305
  140 FORMAT(3X,' AND LESS THAN OR EQUAL TO SATURATED THICKNESS         PL2D306
     1 (',F10.4,A3,')')                                                 PL2D307
  130 WRITE(NO,141)                                                     PL2D308
  141 FORMAT(3X,' -- REENTER',/,'   ?')                                 PL2D309
      READ(NI,56,ERR=130) XF(2)                                         PL2D310
      GO TO 138                                                         PL2D311
  142 IF(XL(2).GE.0.0.AND.XL(2).LE.ST) GO TO 145                        PL2D312
      WRITE(NO,143)                                                     PL2D313
  143 FORMAT(3X,'ZLAST MUST BE GREATER THAN OR EQUAL TO ZERO ')         PL2D314
      IF(IMAGE.GT.1) WRITE(NO,140) ST,IL                                PL2D315
  131 WRITE(NO,141)                                                     PL2D316
      READ(NI,56,ERR=131) XL(2)                                         PL2D317
      GO TO 142                                                         PL2D318
  144 XL(2) = XF(2)                                                     PL2D319
  145 GO TO (720,400),IEDIT                                             PL2D320
C                                                                       PL2D321
C     OBSERVATION TIMES                                                 PL2D322
  720 IF(JFLOW.EQ.2) GO TO 770                                          PL2D323
  724 WRITE(NO,725) IT                                                  PL2D324
  725 FORMAT(3X,'ENTER TFIRST, TLAST, DELTAT  (',A2,')',/,'   ?,?,?')   PL2D325
  730 READ(NI,735,ERR=724) TF,TL,DELT                                   PL2D326
  735 FORMAT(3F10.0)                                                    PL2D327
      DELT = ABS(DELT)                                                  PL2D328
  740 IF(TF.GT.0.0.AND.DELT.LE.1.0E-06) GO TO 760                       PL2D329
      IF(TF.GT.0.0) GO TO 750                                           PL2D330
  744 WRITE(NO,745)                                                     PL2D331
  745 FORMAT(3X,'TFIRST MUST BE GREATER THAN ZERO -- REENTER',/,'   ?') PL2D332
      READ(NI,37,ERR=744) TF                                            PL2D333
      GO TO 740                                                         PL2D334
  750 IF(TL.GT.0.0) GO TO 770                                           PL2D335
  754 WRITE(NO,755)                                                     PL2D336
  755 FORMAT(3X,'TLAST MUST BE GREATER THAN ZERO -- REENTER',/,'   ?')  PL2D337
      READ(NI,37,ERR=754) TL                                            PL2D338
      GO TO 750                                                         PL2D339
  760 TL = TF                                                           PL2D340
  770 GO TO (146,780),IEDIT                                             PL2D341
  780 IF(JFLOW.EQ.2) WRITE(NO,785)                                      PL2D342
  785 FORMAT(3X,'TIME IS NOT A PARAMETER IN STEADY-STATE SOLUTION')     PL2D343
      GO TO 400                                                         PL2D344
C                                                                       PL2D345
C     LIST PROBLEM DEFINITION                                           PL2D346
  146 WRITE(NO,147) NPAGE,(TITLE(I),I=1,30)                             PL2D347
  147 FORMAT(1H1,/,3X,'PLUME2D',/,3X,'VERSION 2.02',                    PL2D348
     1/,3X,'PAGE ',I3,///,3X,30A2,///)                                  PL2D349
      NPAGE = NPAGE + 1                                                 PL2D350
```

```
      IF(IMAGE.GT.1) WRITE(NO,148) IL,ST                                PL2D351
  148 FORMAT(1HO,7X,'SATURATED THICKNESS, (',A2,')  ',24X,F10.4)        PL2D352
      WRITE(NO,149) IL,IT,V,IL,IT,DX,KHAR,IL,IT,DY,P                    PL2D353
  149 FORMAT(8X,'SEEPAGE VELOCITY, (',A2,'/',A2,') ',25X,F10.4,/,       PL2D354
     18X,'X DISPERSION COEFFICIENT (',A2,'**2/',A2,')   ',13X,F10.4,/,  PL2D355
     28X,A1,' DISPERSION COEFFICIENT (',A2,'**2/',A2,')   ',13X,F10.4,/, PL2D356
     38X,'POROSITY ',42X,F10.4)                                         PL2D357
      WRITE(NO,150) RD,IT,DECAY                                         PL2D358
  150 FORMAT(//,8X,'RETARDATION COEFFICIENT',28X,F10.4,/,               PL2D359
     18X,'FIRST ORDER DECAY CONSTANT (1/',A2,')',18X,F10.4)             PL2D360
      GO TO (159,151), JFLOW                                            PL2D361
  151 WRITE(NO,153) KHAR,IL,IL,IM1,IM2,IM3,IL,IT,IL                     PL2D362
  153 FORMAT(//,3X,'STEADY-STATE SOURCE RATES',//,                      PL2D363
     13X,'SOURCE',6X,'X',11X,A1,17X,'RATE',/,                           PL2D364
     25X,'NO',6X,'(',A2,')',8X,'(',A2,')',6X,'(',3A2,                   PL2D365
     3')(CU ',A2,'/',A2,')/',A2,/)                                      PL2D366
      DO 157 I=1,NS                                                     PL2D367
      WRITE(NO,155) I,XS(I),YS(I),Q(I,1)                                PL2D368
  155 FORMAT(5X,I2,F10.2,2X,F10.2,6X,F16.4)                             PL2D369
  157 CONTINUE                                                          PL2D370
      GO TO 171                                                         PL2D371
  159 WRITE(NO,160) IM1,IM2,IM3,IL,IT,IL,IT,IL,KHAR,IL                  PL2D372
  160 FORMAT(//,3X,'SOURCE/RATE SCHEDULE  (',3A2,')(CU ',A2,'/',A2,     PL2D373
     1')/',A2,//,15X,'SOURCE',13X,'RATE',4X,'MASS',8X,'TIME (',A2,')',  PL2D374
     2/,3X,'NO    X (',A2,')    ',A1,' (',A2,')',9X,' NO',5X,'RATE',    PL2D375
     35X,'START',7X,'END',/)                                            PL2D376
      DO 170  I=1,NS                                                    PL2D377
      WRITE(NO,165) I,XS(I),YS(I)                                       PL2D378
  165 FORMAT(/,3X,I2,2F9.2)                                             PL2D379
      NRT = NR(I)                                                       PL2D380
      DO 170  J=1,NRT                                                   PL2D381
         M = J + 1                                                      PL2D382
         WRITE(NO,167) J,Q(I,M),T(I,M-1),T(I,M)                         PL2D383
  167    FORMAT(34X,I2,F12.2,2F9.2)                                     PL2D384
  170 CONTINUE                                                          PL2D385
  171 WRITE(NO,175) IL,XF(1),XL(1),DEL(1),KHAR,XF(2),KHAR,XL(2),KHAR,   PL2D386
     1 DEL(2)                                                           PL2D387
  175 FORMAT(//,8X,'OBSERVATION POINTS (',A2,')',//,                    PL2D388
     112X,'XFIRST =',F10.2,3X,'XLAST =',F10.2,3X,'DELX =',F10.4,/,      PL2D389
     212X,A1,'FIRST =',F10.2,3X,A1,'LAST =',F10.2,3X,'DEL',A1,' =',     PL2D390
     1F10.4)                                                            PL2D391
      IF(JFLOW.EQ.1) WRITE(NO,177) IT,TF,TL,DELT                        PL2D392
  177 FORMAT(/,8X,'OBSERVATION TIMES (',A2,')',//,                      PL2D393
     1 12X,'TFIRST =',F10.2,3X,'TLAST =',F10.2,3X,'DELT =',F10.4)       PL2D394
  180 CONTINUE                                                          PL2D395
      GO TO 400                                                         PL2D396
C                                                                       PL2D397
C                                                                       PL2D398
C                                                                       PL2D399
C ***** SECTION II -- NUMERICAL EVALUATION OF CONCENTRATION AT          PL2D400
C                     SPECIFIED GRID COORDINATES                        PL2D401
C                                                                       PL2D402
C     NUMBER OF OBSERVATION POINTS IN EACH COORDINATE DIRECTION         PL2D403
 2000 CONTINUE                                                          PL2D404
      DO 2020 L=1,2                                                     PL2D405
         NP(L) = 1                                                      PL2D406
         DEL(L) = ABS(DEL(L))                                           PL2D407
         IF(DEL(L).LE.1.0E-03) GO TO 2020                               PL2D408
         DIF = XL(L) - XF(L)                                            PL2D409
         IF(ABS(DIF).LE.1.0E-03) GO TO 2020                             PL2D410
         IF(DIF.LE.0.0) DEL(L)=-DEL(L)                                  PL2D411
         NPTS = ABS(DIF/DEL(L))                                         PL2D412
         REM = DIF - DEL(L)*FLOAT(NPTS)                                 PL2D413
         NPTS = NPTS + 1                                                PL2D414
         NP(L) = NPTS                                                   PL2D415
         IF(ABS(REM).LT.1.0E-03) GO TO 2020                             PL2D416
         NP(L) = NP(L) + 1                                              PL2D417
 2020 CONTINUE                                                          PL2D418
      MAXRW = NP(2)                                                     PL2D419
      MAXCL = NP(1)                                                     PL2D420
```

```
C                                                                       PL2D421
C     TIME COORDINATES                                                  PL2D422
      NTIME = 1                                                         PL2D423
      IF(DELT.LE.1.0E-06) GO TO 2110                                    PL2D424
      NTIME = ABS(TL-TF)/DELT + 1.0                                     PL2D425
      IF(TF.GT.TL) DELT=-DELT                                           PL2D426
 2110 TSOL = TF                                                         PL2D427
      MTIME = NTIME                                                     PL2D428
C                                                                       PL2D429
C                                                                       PL2D430
      DAMK = DX*DECAY*RD/(V*V)                                          PL2D431
      ALPHA=SQRT(1.0+4.0*DAMK)                                          PL2D432
      PE=V/DX                                                           PL2D433
      BETA = DX/DY                                                      PL2D434
      LAMBDA = 1.0/(12.566731*P*SQRT(DX*DY))                            PL2D435
C                                                                       PL2D436
C                                                                       PL2D437
      DO 2660  NT=1,NTIME                                               PL2D438
C                                                                       PL2D439
 2120 LPRT = 1                                                          PL2D440
      LP = 1                                                            PL2D441
      NCFLG = 1                                                         PL2D442
 2140 NROW1 = 1                                                         PL2D443
      NROW2 = MAXROW                                                    PL2D444
 2160 IF(NROW2.GT.MAXRW) NROW2=MAXRW                                    PL2D445
      DO 2580  NROW=NROW1,NROW2                                         PL2D446
      GO TO (2180,2220,2200),NCFLG                                      PL2D447
 2180 NCOL1 = 1                                                         PL2D448
      NCOL2 = MAXCOL                                                    PL2D449
 2200 IF(NCOL2.GT.MAXCL) NCOL2=MAXCL                                    PL2D450
      NCOL = MAXCOL                                                     PL2D451
      IF(NCOL2.EQ.MAXCL) NCOL=NCOL2-NCOL1+1                             PL2D452
 2220 IX1 = NCOL1                                                       PL2D453
      IX2 = NCOL2                                                       PL2D454
C                                                                       PL2D455
      DO 2300  L=1,MAXCOL                                               PL2D456
         CON(L) = 0.0                                                   PL2D457
 2300 CONTINUE                                                          PL2D458
C                                                                       PL2D459
      DO 2440   N=1,NS                                                  PL2D460
         D(1) = ST - YS(N)                                              PL2D461
         IF(ST.GE.0.9E32) D(1)=0.0                                      PL2D462
         D(2) = YS(N)                                                   PL2D463
         D(3) = D(1)                                                    PL2D464
         COEF = 1.0                                                     PL2D465
         IF(D(1).LT.1.0E-03.OR.D(2).LT.1.0E-03) COEF=2.0                PL2D466
         DO 2440  I=IX1,IX2                                             PL2D467
            X = XF(1) + FLOAT(I-1)*DEL(1)                               PL2D468
            IF(I.EQ.NP(1)) X=XL(1)                                      PL2D469
            XXS = X - XS(N)                                             PL2D470
            PEX = PE*XXS                                                PL2D471
            Y = XF(2) + FLOAT(NROW-1)*DEL(2)                            PL2D472
            IF(NROW.EQ.NP(2)) Y=XL(2)                                   PL2D473
            YYS = Y - YS(N)                                             PL2D474
            PEY = PE*YYS                                                PL2D475
            L = I-IX1 + 1                                               PL2D476
            IF(CON(L).LT.0.0) GO TO 2430                                PL2D477
            IF(ABS(XXS).LT.1.0.AND.ABS(YYS).LT.1.0)GO TO 2330           PL2D478
            CALL SOL2D(C,PEX,PEY,TSOL,N,NR(N))                          PL2D479
            CXYT = COEF*C                                               PL2D480
            IF(IMAGE.EQ.1) GO TO 2325                                   PL2D481
C                                                                       PL2D482
            DO 2320  LM=1,2                                             PL2D483
               ZM = ((-1.0)**(LM+1))*YYS                                PL2D484
               IF(D(LM).LT.1.0E-03) GO TO 2320                          PL2D485
               ZIMAGE = 2.0*D(LM) - ZM                                  PL2D486
               PEZ = PE*ZIMAGE                                          PL2D487
               CALL SOL2D(C,PEX,PEZ,TSOL,N,NR(N))                       PL2D488
               CXYT = CXYT + COEF*C                                     PL2D489
               DO 2310  IM = 1,IMAGE                                    PL2D490
```

```
                  ZIMAGE = (2.0*D(LM)+ZM) + 2.0*FLOAT(IM)*D(LM+1)          PL2D491
     1                    + FLOAT(2*IM-2)*D(LM)                            PL2D492
                  PEZ = PE*ZIMAGE                                          PL2D493
                  CALL SOL2D(C,PEX,PEZ,TSOL,N,NR(N))                       PL2D494
                  IF(C.LT.1.0E-06) GO TO 2312                              PL2D495
                  CXYT = CXYT + COEF*C                                     PL2D496
 2310          CONTINUE                                                    PL2D497
 2312          CONTINUE                                                    PL2D498
               DO 2314  IM=1,IMAGE                                         PL2D499
                  ZIMAGE = (2.0*D(LM)-ZM) + 2.0*FLOAT(IM)*D(LM+1)          PL2D500
     1                    + FLOAT(2*IM)*D(LM)                              PL2D501
                  PEZ = PE*ZIMAGE                                          PL2D502
                  CALL SOL2D(C,PEX,PEZ,TSOL,N,NR(N))                       PL2D503
                  IF(C.LT.1.0E-06) GO TO 2320                              PL2D504
                  CXYT = CXYT + COEF*C                                     PL2D505
 2314          CONTINUE                                                    PL2D506
               WRITE(NO,2315) MAXIMG,X,Y                                   PL2D507
 2315          FORMAT(3X,'****** WARNING -- SOLUTION DID NOT',             PL2D508
     1         ' CONVERGE USING',/,9X,I2,' IMAGE WELLS AT X =',            PL2D509
     2         F10.4,'  Z =',F10.4)                                        PL2D510
 2320     CONTINUE                                                         PL2D511
 2325     IF(KFLOW.EQ.1) CXYT=CXYT/2.0                                     PL2D512
          CON(L) = CON(L) + CXYT                                           PL2D513
          GO TO 2340                                                       PL2D514
 2330     CON(L) = -9.9999                                                 PL2D515
 2340     ROW = Y                                                          PL2D516
          COL(L) = X                                                       PL2D517
 2430     CONTINUE                                                         PL2D518
 2440 CONTINUE                                                             PL2D519
C                                                                          PL2D520
C                                                                          PL2D521
C      PRINT CONCENTRATION DISTRIBUTION                                    PL2D522
       GO TO (2460,2560), LPRT                                             PL2D523
 2460 WRITE(NO,147) NPAGE, (TITLE(I),I=1,30)                               PL2D524
       NPAGE = NPAGE + 1                                                  PL2D525
       IF(JFLOW.EQ.2) GO TO 2500                                          PL2D526
       WRITE(NO,2465) TSOL,IT,IM1,IM2,IM3,                                PL2D527
      1(LBL(LP,L),L=1,6),IL                                               PL2D528
 2465 FORMAT(13X,'CONCENTRATION DISTRIBUTION AT ',F10.2,                   PL2D529
      11X,A2,' (',3A2,') ',//,13X,3X,6A2,//,                               PL2D530
      2' *',/,'    * X(',A2,')')                                          PL2D531
       GO TO 2520                                                         PL2D532
 2500 WRITE(NO,2505) IM1,IM2,IM3,(LBL(LP,L),L=1,6),                        PL2D533
      1IL                                                                 PL2D534
 2505 FORMAT(13X,'CONCENTRATION DISTRIBUTION AT STEADY STATE',             PL2D535
      1' (',3A2,') ',//13X,3X,6A2,//,                                     PL2D536
      2' *',/,'    * X(',A2,')')                                          PL2D537
 2520 CONTINUE                                                             PL2D538
       WRITE(NO,2525) (COL(L),L=1,NCOL)                                   PL2D539
 2525 FORMAT('      *',4X,7F10.2)                                          PL2D540
       WRITE(NO,2545) KHAR,IL                                             PL2D541
 2545 FORMAT(1X,A1,'(',A2,') *',/,9X,'*')                                  PL2D542
C                                                                          PL2D543
 2560 WRITE(NO,2565) ROW,(CON(L),L=1,NCOL)                                 PL2D544
 2565 FORMAT(2X,F8.2,7F10.4)                                               PL2D545
       LPRT = 2                                                           PL2D546
 2580 CONTINUE                                                             PL2D547
       IF(NROW2.EQ.MAXRW) GO TO 2600                                      PL2D548
       NROW1 = NROW1 + MAXROW                                             PL2D549
       NROW2 = NROW2 + MAXROW                                             PL2D550
       LPRT = 1                                                           PL2D551
       LP = 2                                                             PL2D552
       NCFLG = 2                                                          PL2D553
       GO TO 2160                                                         PL2D554
 2600 IF(NCOL2.EQ.MAXCL) GO TO 2640                                        PL2D555
       NCOL1 = NCOL1 + MAXCOL                                             PL2D556
       NCOL2 = NCOL2 + MAXCOL                                             PL2D557
       LPRT = 1                                                           PL2D558
       LP = 2                                                             PL2D559
       NCFLG = 3                                                          PL2D560
```

```
      GO TO 2140                                                        PL2D561
 2640 CONTINUE                                                          PL2D562
      TSOL = TSOL + DELT                                                PL2D563
      IF(NT.EQ.MTIME) TSOL=TL                                           PL2D564
 2660 CONTINUE                                                          PL2D565
C ***** SECTION III -- PROBLEM REDEFINITION AND CONTROL OF EXECUTION     PL2D566
C                                                                       PL2D567
C                                                                       PL2D568
  400 CONTINUE                                                          PL2D569
      IF(IEDIT.EQ.2)GO TO 401                                           PL2D570
      WRITE(NO,1001)KHAR,KHAR,KHAR,KHAR                                 PL2D571
      IEDIT = 2                                                         PL2D572
  401 KNTL = 0                                                          PL2D573
      WRITE(NO,405)                                                     PL2D574
  405 FORMAT(//,3X,'ENTER NEXT COMMAND',/,'   ?')                       PL2D575
  410 READ(NI,415) NEXT                                                 PL2D576
  415 FORMAT(A2)                                                        PL2D577
C                                                                       PL2D578
      DO 420  I=1,20                                                    PL2D579
         IF(NEXT.EQ.IC(I)) GO TO 430                                    PL2D580
  420 CONTINUE                                                          PL2D581
      WRITE(NO,425)                                                     PL2D582
  425 FORMAT(3X,'ERROR IN LAST COMMAND -- REENTER',/,'   ?')            PL2D583
      GO TO 410                                                         PL2D584
  430 GO TO (91,60,70,80,86,86,50,450,124,133,133,3060,1,               PL2D585
     12000,700,146,1000,602,1320,720),I                                 PL2D586
C                                                                       PL2D587
C     NEW SET OF X AND Y OBSERVATIONS                                   PL2D588
  450 KNTL = 1                                                          PL2D589
      GO TO 124                                                         PL2D590
C                                                                       PL2D591
C                                                                       PL2D592
C     NEW SOURCE/RATE SCHEDULE                                          PL2D593
 3060 WRITE(NO,3065)                                                    PL2D594
 3065 FORMAT(3X,'ADD(A),DELETE(D),MODIFY(M) A SOURCE OR RETURN(R)'      PL2D595
     1' TO EDIT ?')                                                     PL2D596
 3070 READ(NI,3075) ISK                                                 PL2D597
 3075 FORMAT(A1)                                                        PL2D598
      DO 3080  I=1,4                                                    PL2D599
         IF(ISK.EQ.IS(I)) GO TO 3090                                    PL2D600
 3080 CONTINUE                                                          PL2D601
      WRITE(NO,3085)                                                    PL2D602
 3085 FORMAT(3X,'ERROR IN SELECTION -- REENTER ?')                      PL2D603
      GO TO 3070                                                        PL2D604
 3090 GO TO (400,3100,3450,3490),I                                      PL2D605
C                                                                       PL2D606
C     MODIFY SOURCE                                                     PL2D607
C                                                                       PL2D608
 3100 WRITE(NO,3105) NS                                                 PL2D609
 3105 FORMAT(3X,I2,' SOURCES IN CURRENT SCHEDULE',/,                    PL2D610
     13X,'ENTER SOURCE TO MODIFY',/,'   ?')                            PL2D611
      READ(NI,1465,ERR=3100) FDUM                                       PL2D612
      JS=FDUM                                                           PL2D613
      IF(JS.GT.0.AND.JS.LE.NS) GO TO 3220                               PL2D614
      WRITE(NO,3215) JS                                                 PL2D615
 3215 FORMAT(3X,'SOURCE',I4,' NOT IN SCHEDULE')                         PL2D616
      GO TO 3060                                                        PL2D617
 3220 GO TO (3230,3260),JFLOW                                           PL2D618
 3230 WRITE(NO,3235) JS,XS(JS),IL,KHAR,YS(JS),IL,IT,                    PL2D619
     1IM1,IM2,IM3,IL,IT,IL                                              PL2D620
 3235 FORMAT(3X,'SOURCE ',I2,':   X =',F8.2,A3,2X,',',A1,' =',          PL2D621
     1F8.2,A3,//,3X,'RATE',7X,'MASS RATE',14X,'TIME (',                 PL2D622
     2A2,')',/,4X,'NO',3X,'(',3A2,')(CU ',A2,'/',A2,')/',A2,            PL2D623
     35X,'START',7X,'END',/)                                            PL2D624
      NRT = NR(JS)                                                      PL2D625
      DO 3250  J=1,NRT                                                  PL2D626
         M = J + 1                                                      PL2D627
         WRITE(NO,3245) J,Q(JS,M),T(JS,M-1),T(JS,M)                     PL2D628
 3245    FORMAT(4X,I2,5X,F14.2,7X,F8.3,3X,F8.2)                         PL2D629
 3250 CONTINUE                                                          PL2D630
```

```
      GO TO 3270                                                        PL2D631
 3260 WRITE(NO,3265) JS,XS(JS),IL,KHAR,YS(JS),IL,Q(JS,1),               PL2D632
     1IM1,IM2,IM3,IL,IT,IL                                              PL2D633
 3265 FORMAT(3X,'SOURCE ',I2,':   X =',F8.2,A3,2X,',',A1,' =',          PL2D634
     1F8.2,A3,/,3X,'STEADY-STATE MASS RATE =',F16.4,                    PL2D635
     2' (',3A3,')(CU ',A2,'/',A2,')/',A2,/)                             PL2D636
 3270 WRITE(NO,3275)                                                    PL2D637
 3275 FORMAT(3X,'CHANGE COORDINATES (Y/N)?')                            PL2D638
      READ(NI,3075) JC                                                  PL2D639
      IF(JC.NE.IY) GO TO 3290                                           PL2D640
      IF(KFLOW.EQ.2) GO TO 3277                                         PL2D641
 3276 WRITE(NO,1410) JS,IL                                              PL2D642
      READ(NI,1425,ERR=3276) XS(JS),YS(JS)                              PL2D643
      GO TO 3290                                                        PL2D644
 3277 WRITE(NO,1415) JS,IL                                              PL2D645
      READ(NI,1425,ERR=3277) XS(JS),YS(JS)                              PL2D646
 3280 IF(YS(JS).GE.0.0.AND.YS(JS).LE.ST) GO TO 3290                     PL2D647
 3281 WRITE(NO,1435) ST,IL                                              PL2D648
      READ(NI,37,ERR=3281) YS(JS)                                       PL2D649
      GO TO 3280                                                        PL2D650
 3290 GO TO (3300,3430),JFLOW                                           PL2D651
C                                                                       PL2D652
C     TRANSIENT SOURCES                                                 PL2D653
 3300 WRITE(NO,3305) JS                                                 PL2D654
 3305 FORMAT(3X,'MODIFY RATE SCHEDULE FOR SOURCE',I3,' (Y/N) ?')        PL2D655
      READ(NI,3075) JY                                                  PL2D656
      IF(JY.NE.IY) GO TO 3060                                           PL2D657
 3310 WRITE(NO,3315)                                                    PL2D658
 3315 FORMAT(3X,'ENTER RATE TO BE CHANGED',/,                           PL2D659
     13X,'(ENTER 0 TO CHANGE ALL RATES)',/,'   ?')                      PL2D660
      READ(NI,1465,ERR=3310) FDUM                                       PL2D661
      JR=FDUM                                                           PL2D662
      IF(JR.LE.0) GO TO 3350                                            PL2D663
      IF(JR.LE.NR(JS)) GO TO 3330                                       PL2D664
      WRITE(NO,3325) JR                                                 PL2D665
 3325 FORMAT(3X,'RATE ',I2,' NOT IN CURRENT SCHEDULE')                  PL2D666
      GO TO 3300                                                        PL2D667
 3330 WRITE(NO,3335) JS,JR,T(JS,JR),T(JS,JR+1),IT                       PL2D668
 3335 FORMAT(3X,'SOURCE ',I2,', RATE ',I2,' STARTS AT',F8.2,            PL2D669
     1' AND ENDS AT',F8.2,A3,/,3X,'ENTER NEW MASS RATE',/,'   ?')       PL2D670
      M = JR + 1                                                        PL2D671
      READ(NI,3345,ERR=3330) Q(JS,M)                                    PL2D672
 3345 FORMAT(F10.0)                                                     PL2D673
      GO TO 3300                                                        PL2D674
C                                                                       PL2D675
 3350 NRT = NR(JS)                                                      PL2D676
      DO 3360  J=1,NRT                                                  PL2D677
         M = J + 1                                                      PL2D678
         Q(JS,M) = 0.0                                                  PL2D679
         T(JS,M) = 0.0                                                  PL2D680
 3360 CONTINUE                                                          PL2D681
 3370 WRITE(NO,1455) JS,MAXRT                                           PL2D682
 3380 READ(NI,1465,ERR=3370) FDUM                                       PL2D683
      NR(JS)=FDUM                                                       PL2D684
      IF(NR(JS).GT.0.AND.NR(JS).LE.MAXRT) GO TO 3390                    PL2D685
      WRITE(NO,1475) MAXRT                                              PL2D686
      GO TO 3380                                                        PL2D687
 3390 CONTINUE                                                          PL2D688
      NRT = NR(JS)                                                      PL2D689
      DO  3420  J=1,NRT                                                 PL2D690
         M = J + 1                                                      PL2D691
 3394    WRITE(NO,1485) JS,J,T(JS,M-1),IT                               PL2D692
         READ(NI,1495,ERR=3394) Q(JS,M),T(JS,M)                         PL2D693
 3400    IF(T(JS,M).GT.T(JS,M-1)) GO TO 3410                            PL2D694
 3404    WRITE(NO,1505)                                                 PL2D695
         READ(NI,37,ERR=3404) T(JS,M)                                   PL2D696
         GO TO 3400                                                     PL2D697
 3410    CONTINUE                                                       PL2D698
 3420 CONTINUE                                                          PL2D699
      GO TO 3060                                                        PL2D700
```

```
C                                                                       PL2D701
C     STEADY-STATE SOURCES                                              PL2D702
 3430 WRITE(NO,3435) JS                                                 PL2D703
 3435 FORMAT(3X,'CHANGE STEADY-STATE RATE FOR SOURCE ',I2,' (Y/N) ?')   PL2D704
      READ(NI,3075) JC                                                  PL2D705
      IF(JC.NE.IY) GO TO 3060                                           PL2D706
 3444 WRITE(NO,3445) JS                                                 PL2D707
 3445 FORMAT(3X,'ENTER NEW STEADY-STATE MASS RATE FOR SOURCE ',I2,/,    PL2D708
     1'   ?')                                                           PL2D709
      READ(NI,3345,ERR=3444) Q(JS,1)                                    PL2D710
      GO TO 3060                                                        PL2D711
C                                                                       PL2D712
C     ADD A NEW SOURCE                                                  PL2D713
C                                                                       PL2D714
 3450 NS = NS + 1                                                       PL2D715
      JS = NS                                                           PL2D716
      IF(KFLOW.EQ.2) GO TO 3455                                         PL2D717
 3454 WRITE(NO,1410) JS,IL                                              PL2D718
      READ(NI,1425,ERR=3454) XS(JS),YS(JS)                              PL2D719
      GO TO 3470                                                        PL2D720
 3455 WRITE(NO,1415) JS,IL                                              PL2D721
      READ(NI,1425,ERR=3455) XS(JS),YS(JS)                              PL2D722
 3460 IF(YS(JS).GE.O.O.AND.YS(JS).LE.ST) GO TO 3470                     PL2D723
 3464 WRITE(NO,1435) ST,IL                                              PL2D724
      READ(NI,37,ERR=3464) YS(JS)                                       PL2D725
      GO TO 3460                                                        PL2D726
 3470 GO TO (3370,3480),JFLOW                                           PL2D727
C                                                                       PL2D728
C     STEADY-STATE SOURCES                                              PL2D729
 3480 WRITE(NO,3485) JS                                                 PL2D730
 3485 FORMAT(3X,'ENTER STEADY-STATE MASS RATE FOR SOURCE ',I2,          PL2D731
     1/,'   ?')                                                        PL2D732
      READ(NI,3345,ERR=3480) Q(JS,1)                                    PL2D733
      NR(JS) = O                                                        PL2D734
      GO TO 3060                                                        PL2D735
C                                                                       PL2D736
C     DELETE A SOURCE                                                   PL2D737
C                                                                       PL2D738
 3490 IF(NS.GT.1) GO TO 3500                                            PL2D739
      WRITE(NO,3495)                                                    PL2D740
 3495 FORMAT(3X,'ONLY ONE SOURCE IN SCHEDULE -- CAN NOT DELETE',/)      PL2D741
      GO TO 3060                                                        PL2D742
 3500 WRITE(NO,3505) IL,IL,IL                                           PL2D743
 3505 FORMAT(3X,'SOURCE',6X,'X (',A2,')',3X,'Y (',A2,')',3X,            PL2D744
     1'Z (',A2,')',/)                                                  PL2D745
      DO 3520  I=1,NS                                                   PL2D746
         WRITE(NO,3515) I,XS(I),YS(I)                                   PL2D747
 3515    FORMAT(5X,I2,3X,F8.2,3X,F8.2)                                  PL2D748
 3520 CONTINUE                                                          PL2D749
 3530 WRITE(NO,3535)                                                    PL2D750
 3535 FORMAT(3X,'ENTER SOURCE TO DELETE',/,                             PL2D751
     13X,'(ENTER O TO CANCEL)',/,'   ?')                               PL2D752
      READ(NI,1465,ERR=3530) FDUM                                       PL2D753
      JS=FDUM                                                           PL2D754
      IF(JS.LE.O) GO TO 3060                                            PL2D755
      IF(JS.LE.NS) GO TO 3550                                           PL2D756
      WRITE(NO,3545) JS                                                 PL2D757
 3545 FORMAT(3X,'SOURCE ',I2,' NOT IN CURRENT SCHEDULE')                PL2D758
      GO TO 3530                                                        PL2D759
 3550 WRITE(NO,3555) JS                                                 PL2D760
 3555 FORMAT(3X,'DELETE SOURCE ',I2,' (Y/N)?')                          PL2D761
      READ(NI,3075) JC                                                  PL2D762
      IF(JC.NE.IY) GO TO 3530                                           PL2D763
      NSD = NS - 1                                                      PL2D764
      GO TO (3560,3590),JFLOW                                           PL2D765
C                                                                       PL2D766
C     TRANSIENT SOURCES                                                 PL2D767
 3560 IF(JS.EQ.NS) GO TO 3575                                           PL2D768
      DO  3570  J=JS,NSD                                                PL2D769
         XS(J) = XS(J+1)                                                PL2D770
```

```
          YS(J) = YS(J+1)                                                    PL2D771
          NR(J) = NR(J+1)                                                    PL2D772
          NRT = NR(J)                                                        PL2D773
             DO 3570  K=1,NRT                                                PL2D774
             M = K + 1                                                       PL2D775
             Q(J,M) = Q(J+1,M)                                               PL2D776
             T(J,M) = T(J+1,M)                                               PL2D777
 3570 CONTINUE                                                               PL2D778
 3575 NRT = NR(NS)                                                           PL2D779
      DO 3580  K=1,NRT                                                       PL2D780
         M = K + 1                                                           PL2D781
         Q(NS,M) = 0.0                                                       PL2D782
         T(NS,M) = 0.0                                                       PL2D783
 3580 CONTINUE                                                               PL2D784
      NR(NS) = 0                                                             PL2D785
      NS = NSD                                                               PL2D786
      GO TO 3060                                                             PL2D787
C                                                                            PL2D788
C     STEADY-STATE SOURCES                                                   PL2D789
 3590 IF(JS.EQ.NS) GO TO 3605                                                PL2D790
      DO 3600  J=JS,NSD                                                      PL2D791
         Q(J,1) = Q(J+1,1)                                                   PL2D792
         XS(J) = XS(J+1)                                                     PL2D793
         YS(J) = YS(J+1)                                                     PL2D794
 3600 CONTINUE                                                               PL2D795
 3605 Q(NS,1) = 0.0                                                          PL2D796
      NS = NSD                                                               PL2D797
      GO TO 3060                                                             PL2D798
C                                                                            PL2D799
C     CONFIRM WHETHER SATURATED THICKNESS IS A VARIABLE                      PL2D800
  602 IF(KFLOW.EQ.2)GO TO 30                                                 PL2D801
      WRITE(NO,605)                                                          PL2D802
  605 FORMAT(3X,'SATURATED THICKNESS IS NOT A VARIABLE IN',/,                PL2D803
     13X,'X-Y COORIDINATE SYSTEM (VERTICALLY AVERAGED SOLUTION)')            PL2D804
      GO TO 400                                                              PL2D805
C                                                                            PL2D806
C     MENU OF EDIT COMMANDS FOR PLUMES VERSION 2.02                          PL2D807
 1000 WRITE(NO,1001)KHAR,KHAR,KHAR,KHAR                                      PL2D808
 1001 FORMAT(1H1,/,3X,'MENU OF EDIT COMMANDS',//,                            PL2D809
     1'  RETARDATION COEFFICIENT  RD      OBSERVATION POINTS  OB',/,          PL2D810
     2'  POROSITY                 PO      X COORDINATES       XC',/,          PL2D811
     3'  SEEPAGE VELOCITY         VX',6X,A1,' COORDINATES',7X,A1'C',/,        PL2D812
     4'  X DISPERSION COEFFICIENT DX      MENU OF COMMANDS    MU',/,          PL2D813
     52X,A1,' DISPERSION COEFFICIENT D',A1,6X,'LIST INPUT DATA     LI',       PL2D814
     6/,'  DECAY CONSTANT          DE      RUN CALCULATIONS    RN',/,          PL2D815
     7'  SOURCE RATE SCHEDULE     RT      DONE                DN',/,          PL2D816
     8'  NEW PROBLEM              NP      SATURATED THICKNESS ST',/,          PL2D817
     9'  CHANGE SOLUTION/SOURCES  CS      OBSERVATION TIMES   TC')            PL2D818
      GO TO 400                                                              PL2D819
C                                                                            PL2D820
  700 STOP                                                                   PL2D821
      END                                                                    PL2D822
```

APPENDIX E

Listing of Utility Function Subroutines

```
      FUNCTION BIO(Z)                                                   BIO 001
C     JAN WAGNER                                                        BIO 002
C     SCHOOL OF CHEMICAL ENGINEERING                                    BIO 003
C     OKLAHOMA STATE UNIVERSITY                                         BIO 004
C     STILLWATER, OK  74078                                             BIO 005
C     TELEPHONE: (405) 624-5280                                         BIO 006
C                                                                       BIO 007
C     REVISED: 6 JANUARY 1983                                           BIO 008
C                                                                       BIO 009
C     EVALUATION OF MODIFIED BESSEL FUNCTION OF THE FIRST KIND          BIO 010
C     OF ORDER ZERO                                                     BIO 011
C     POLYNOMIAL APPROXIMATIONS ARE USED FOR IO(Z)                      BIO 012
C     SEE SECTION 9.8 OF ABRAMOWITZ AND STEGUN (1966)                   BIO 013
      DIMENSION A(9)                                                    BIO 014
      COMMON/IO/NI,NO                                                   BIO 015
      DATA A/0.9189385,4.32105045,6.09540829,6.45308739,4.6926023,      BIO 016
     13.88357842,3.63608323,4.10583047,5.540702353/                     BIO 017
C                                                                       BIO 018
      IF(Z.LE.0.0) GO TO 200                                            BIO 019
      T = Z/3.75                                                        BIO 020
      IF(Z.GT.3.75) GO TO 100                                           BIO 021
      T2 = T*T                                                          BIO 022
      T4 = T2*T2                                                        BIO 023
      T6 = T2*T4                                                        BIO 024
      T8 = T2*T6                                                        BIO 025
      T10 = T2*T8                                                       BIO 026
      T12 = T2*T10                                                      BIO 027
      BIO = 1.0 + 3.5156229*T2 + 3.0899424*T4 + 1.2067492*T6            BIO 028
     1  + 0.2659732*T8 + 0.0360768*T10+ 0.0045813*T12                   BIO 029
      RETURN                                                            BIO 030
  100 CONTINUE                                                          BIO 031
      SUM = 0.0                                                         BIO 032
      DO 150  I=1,9                                                     BIO 033
      SIGN = (-1.0)**(I+1)                                              BIO 034
      IF(I.EQ.2) SIGN=1.0                                               BIO 035
      ARG = -1.0*A(I) - 0.5*ALOG(Z) - FLOAT(I-1)*ALOG(T)                BIO 036
      SUM = SUM + SIGN*EXP(ARG)                                         BIO 037
  150 CONTINUE                                                          BIO 038
      BIOLOG = ALOG(SUM) + Z                                            BIO 039
      BIO = EXP(BIOLOG)                                                 BIO 040
      RETURN                                                            BIO 041
  200 CONTINUE                                                          BIO 042
      IF(Z.LT.0.0) GO TO 300                                            BIO 043
      BIO = 1.0                                                         BIO 044
      RETURN                                                            BIO 045
  300 WRITE(NO,305) Z                                                   BIO 046
  305 FORMAT(6X,'ARGUMENT OF BESSEL FUNCTION IO(Z) IS NEGATIVE',/,      BIO 047
     16X,'Z = ',E12.6,'  -- PROGRAM TERMINATED')                        BIO 048
      STOP                                                              BIO 049
      END                                                               BIO 050
```

```
      FUNCTION BIOLOG(Z)                                                BIOL001
C     JAN WAGNER                                                        BIOL002
C     SCHOOL OF CHEMICAL ENGINEERING                                    BIOL003
C     OKLAHOMA STATE UNIVERSITY                                         BIOL004
C     STILLWATER, OK  74078                                             BIOL005
C     TELEPHONE: (405) 624-5280                                         BIOL006
C                                                                       BIOL007
C     REVISED: 6 JANUARY 1983                                           BIOL008
C                                                                       BIOL009
C     EVALUATION OF THE NATURAL LOG OF A MODIFIED BESSEL                BIOL010
C     FUNCTION OF THE FIRST KIND OF ORDER ZERO                          BIOL011
C     POLYNOMIAL APPROXIMATIONS ARE USED FOR IO(Z)                      BIOL012
C     SEE SECTION 9.8 OF ABRAMOWITZ AND STEGUN (1966)                   BIOL013
      DIMENSION A(9)                                                    BIOL014
      COMMON/IO/NI,NO                                                   BIOL015
      DATA A/0.9189385,4.32105045,6.09540829,6.45308739,4.6926023,      BIOL016
     13.88357842,3.63608323,4.10583047,5.540702353/                     BIOL017
C                                                                       BIOL018
      IF(Z.LE.0.0) GO TO 200                                            BIOL019
      T = Z/3.75                                                        BIOL020
      IF(Z.GT.3.75) GO TO 100                                           BIOL021
      T2 = T*T                                                          BIOL022
      T4 = T2*T2                                                        BIOL023
      T6 = T2*T4                                                        BIOL024
      T8 = T2*T6                                                        BIOL025
      T10 = T2*T8                                                       BIOL026
      T12 = T2*T10                                                      BIOL027
      BIO = 1.0 + 3.5156229*T2 + 3.0899424*T4 + 1.2067492*T6            BIOL028
     1           + 0.2659732*T8 + 0.0360768*T10 + 0.0045813*T12         BIOL029
      BIOLOG = ALOG(BIO)                                                BIOL030
      RETURN                                                            BIOL031
  100 CONTINUE                                                          BIOL032
      SUM = 0.0                                                         BIOL033
      DO 150  I=1,9                                                     BIOL034
      SIGN = (-1.0)**(I+1)                                              BIOL035
      IF(I.EQ.2) SIGN=1.0                                               BIOL036
      ARG = -1.0*A(I) - 0.5*ALOG(Z) - FLOAT(I-1)*ALOG(T)                BIOL037
      SUM = SUM + SIGN*EXP(ARG)                                         BIOL038
  150 CONTINUE                                                          BIOL039
      BIOLOG = ALOG(SUM) + Z                                            BIOL040
      RETURN                                                            BIOL041
  200 CONTINUE                                                          BIOL042
      IF(Z.LT.0.0) GO TO 300                                            BIOL043
      BIO = 1.0                                                         BIOL044
      RETURN                                                            BIOL045
  300 WRITE(NO,305) Z                                                   BIOL046
  305 FORMAT(6X,'ARGUMENT OF BESSEL FUNCTION IO(Z) IS NEGATIVE'./,      BIOL047
     16X,'Z = ',E12.6,'  -- PROGRAM TERMINATED')                        BIOL048
      STOP                                                              BIOL049
      END                                                               BIOL050
```

```
      FUNCTION BKO(Z)                                                    BKO 001
C     JAN WAGNER                                                         BKO 002
C     SCHOOL OF CHEMICAL ENGINEERING                                     BKO 003
C     OKLAHOMA STATE UNIVERSITY                                          BKO 004
C     STILLWATER, OK  74078                                              BKO 005
C     TELEPHONE: (405) 624-5280                                          BKO 006
C                                                                        BKO 007
C     REVISED: 6 JANUARY 1983                                            BKO 008
C                                                                        BKO 009
C     EVALUATION OF MODIFIED BESSEL FUNTION OF SECOND KIND               BKO 010
C     OF ORDER ZERO                                                      BKO 011
C     POLYNOMIAL APPROXIMATIONS ARE USED FOR KO(Z)                       BKO 012
C     SEE SECTION 9.8 OF ABRAMOWITZ AND STEGUN (1966)                    BKO 013
      COMMON/IO/NI,NO                                                    BKO 014
C                                                                        BKO 015
      IF(Z.LE.0.0) GO TO 200                                             BKO 016
      T = Z/2.0                                                          BKO 017
      T2 = T*T                                                           BKO 018
      T4 = T2*T2                                                         BKO 019
      T6 = T2*T4                                                         BKO 020
      IF(Z.GT.2.0) GO TO 100                                             BKO 021
      T8 = T2*T6                                                         BKO 022
      T10 = T2*T8                                                        BKO 023
      T12 = T2*T10                                                       BKO 024
      BKO = -1.0*ALOG(T)*BIO(Z) - 0.57721566                             BKO 025
     1     + 0.42278420*T2 + 0.23069756*T4 + 0.03488590*T6               BKO 026
     2     + 0.00262698*T8 + 0.00010750*T10+ 0.0740E-04*T12              BKO 027
      RETURN                                                             BKO 028
  100 CONTINUE                                                           BKO 029
      SUM = (1.25331414 - 0.07832358/T + 0.02189568/T2                   BKO 030
     1     - 0.01062446/(T*T2) + 0.00587872/T4                           BKO 031
     2     - 0.00251540/(T*T4) + 0.00053208/T6)                          BKO 032
      BKOLOG = ALOG(SUM) - Z - 0.5*ALOG(Z)                               BKO 033
      BKO = EXP(BKOLOG)                                                  BKO 034
      RETURN                                                             BKO 035
  200 CONTINUE                                                           BKO 036
      WRITE(NO,205) Z                                                    BKO 037
  205 FORMAT(6X,'ARGUMENT OF BESSEL FUNCTION KO(Z) IS LESS THAN',        BKO 038
     1' OR EQUAL TO ZERO',/,6X,'Z = ',E12.6,'  -- PROGRAM TERMINATED')   BKO 039
      STOP                                                               BKO 040
      END                                                                BKO 041
```

```
      FUNCTION BKOLOG(Z)                                                BKOL001
C     JAN WAGNER                                                        BKOL002
C     SCHOOL OF CHEMICAL ENGINEERING                                    BKOL003
C     OKLAHOMA STATE UNIVERSITY                                         BKOL004
C     STILLWATER, OK  74078                                             BKOL005
C     TELEPHONE: (405) 624-5280                                         BKOL006
C                                                                       BKOL007
C     REVISED: 6 JANUARY 1983                                           BKOL008
C                                                                       BKOL009
C     NATURAL LOG OF MODIFIED BESSEL FUNTION OF SECOND KIND             BKOL010
C     OF ORDER ZERO                                                     BKOL011
C     POLYNOMIAL APPROXIMATIONS ARE USED FOR KO(Z)                      BKOL012
C     SEE SECTION 9.8 OF ABRAMOWITZ AND STEGUN (1966)                   BKOL013
      COMMON/IO/NI,NO                                                   BKOL014
C                                                                       BKOL015
      IF(Z.LE.0.0) GO TO 200                                            BKOL016
      T = Z/2.0                                                         BKOL017
      T2 = T*T                                                          BKOL018
      T4 = T2*T2                                                        BKOL019
      T6 = T2*T4                                                        BKOL020
      IF(Z.GT.2.0) GO TO 100                                            BKOL021
      T8 = T2*T6                                                        BKOL022
      T10 = T2*T8                                                       BKOL023
      T12 = T2*T10                                                      BKOL024
      BKO = -1.0*ALOG(T)*BIO(Z) - 0.57721566                            BKOL025
     1      + 0.42278420*T2 + 0.23069756*T4 + 0.03488590*T6            BKOL026
     2      + 0.00262698*T8 + 0.00010750*T10+ 0.0740E-04*T12           BKOL027
      BKOLOG = ALOG(BKO)                                                BKOL028
      RETURN                                                            BKOL029
  100 CONTINUE                                                          BKOL030
      SUM = (1.25331414 - 0.07832358/T + 0.02189568/T2                  BKOL031
     1      - 0.01062446/(T*T2) + 0.00587872/T4                        BKOL032
     2      - 0.00251540/(T*T4) + 0.00053208/T6)                       BKOL033
      BKOLOG = ALOG(SUM) - Z - 0.5*ALOG(Z)                              BKOL034
      RETURN                                                            BKOL035
  200 CONTINUE                                                          BKOL036
      WRITE(NO,205) Z                                                   BKOL037
  205 FORMAT(6X,'ARGUMENT OF BESSEL FUNCTION KO(Z) IS LESS THAN',       BKOL038
     1' OR EQUAL TO ZERO',/,6X,'Z = ',E12.6,'  -- PROGRAM TERMINATED')  BKOL039
      STOP                                                              BKOL040
      END                                                               BKOL041
```

```
      FUNCTION ERFC(Z)                                                  ERFC001
C     JAN WAGNER                                                        ERFC002
C     SCHOOL OF CHEMICAL ENGINEERING                                    ERFC003
C     OKLAHOMA STATE UNIVERSITY                                         ERFC004
C     STILLWATER, OK  74078                                             ERFC005
C     TELEPHONE: (405) 624-5280                                         ERFC006
C                                                                       ERFC007
C     REVISED: 6 JANUARY 1983                                           ERFC008
C                                                                       ERFC009
C     RATIONAL APPROXIMATION OF THE COMPLIMENTARY ERROR FUNCTION        ERFC010
C     SEE SECTION 7.1 OF ABRAMOWITZ AND STEGUN (1966)                   ERFC011
C     THE FOLLOWING IDENTITIES ARE USED TO HANDLE NEGATIVE ARGUMENTS    ERFC012
C      ERFC( Z) = 1 - ERF(Z)                                            ERFC013
C      ERF (-Z) = -ERF (Z)                                              ERFC014
C                                                                       ERFC015
      REAL*8 COEFLG,DERFC,DI,FX,TERMI,TERMO,SUM,X                       ERFC016
      COMMON/IO/NI,NO                                                   ERFC017
C                                                                       ERFC018
      X = ABS(Z)                                                        ERFC019
      IF (X.GT.3.0D00) GO TO 50                                         ERFC020
C                                                                       ERFC021
C     FOR X<3 A RATIONAL APPROXIMATION OF THE COMPLIMENTARY ERROR       ERFC022
C     FUNCTION IS USED.                                                 ERFC023
C                                                                       ERFC024
      DERFC = 1.0D00/((1.0D00 + 7.05230784D-02*X + 4.22820123D-02*(X**2) ERFC025
     1    + 9.2705272D-03*(X**3) + 1.520143D-04*(X**4)                  ERFC026
     2    + 2.76572D-04*(X**5) + 4.30638D-05*(X**6))**16)               ERFC027
      GO TO 100                                                         ERFC028
C                                                                       ERFC029
C     FOR X>3 AN ASYMPTOTIC EXPANSION OF THE COMPLIMENTARY ERROR        ERFC030
C     FUNCTION IS USED.                                                 ERFC031
   50 COEFLG = X*X + DLOG(X) + 0.57236494D00                            ERFC032
      FX = 2.0D00*X*X                                                   ERFC033
      SUM = 1.0D00                                                      ERFC034
      TERMO = 1.0D00                                                    ERFC035
      DO 60 I=2,50                                                      ERFC036
      DI = I                                                            ERFC037
      TERMI = -TERMO*(2.0D00*DI - 3.0D00)/FX                            ERFC038
      IF(DABS(TERMI).GT.DABS(TERMO)) GO TO 70                           ERFC039
      SUM = SUM + TERMI                                                 ERFC040
      TEST = TERMI/SUM                                                  ERFC041
      IF(ABS(TEST).LT.1.0E-16) GO TO 70                                 ERFC042
      TERMO = TERMI                                                     ERFC043
   60 CONTINUE                                                          ERFC044
      WRITE(NO,65)                                                      ERFC045
   65 FORMAT(6X,'*** WARNING -- ASYMPTOTIC EXPANSION FOR ERFC DID NOT', ERFC046
     1'                          CONVERGE WITH 50 TERMS IN THE SUMMATION') ERFC047
   70 SUM = DLOG(SUM) - COEFLG                                          ERFC048
      IF(SUM.LT.-72.0D00) SUM=-72.0D00                                  ERFC049
      DERFC = DEXP(SUM)                                                 ERFC050
  100 CONTINUE                                                          ERFC051
C                                                                       ERFC052
C     FOR Z<0, ERFC(-Z) = 2-ERFC(Z)                                     ERFC053
      ERFC = DERFC                                                      ERFC054
      IF(Z.LT.0.0) ERFC=2.0D00-DERFC                                    ERFC055
      RETURN                                                            ERFC056
      END                                                               ERFC057
```

```
      FUNCTION E1LOG(Z)                                                 E1LG001
C     JAN WAGNER                                                        E1LG002
C     SCHOOL OF CHEMICAL ENGINEERING                                    E1LG003
C     OKLAHOMA STATE UNIVERSITY                                         E1LG004
C     STILLWATER, OK  74078                                             E1LG005
C     TELEPHONE: (405) 624-5280                                         E1LG006
C                                                                       E1LG007
C     REVISED: 6 JANUARY 1983                                           E1LG008
C                                                                       E1LG009
C     EVALUATION OF THE NATURAL LOG OF THE EXPONENTIAL INTEGRAL         E1LG010
C     POLYNOMIAL APPROXIMATIONS ARE USED FOR E1(Z)                      E1LG011
C     SEE SECTION 5.1 OF ABRAMOWITZ AND STEGUN (1966)                   E1LG012
      COMMON/IO/NI,NO                                                   E1LG013
C                                                                       E1LG014
      IF(Z.LE.0.0) GO TO 200                                            E1LG015
      Z2 = Z*Z                                                          E1LG016
      Z3 = Z*Z2                                                         E1LG017
      IF(Z.GT.1.0) GO TO 100                                            E1LG018
C     ARGUMENTS LESS THAN UNITY                                         E1LG019
      E1 = - 0.57721566     + 0.99999193*Z      - 0.24991055*Z2         E1LG020
     1     + 0.05519968*Z3 - 0.00976004*Z2*Z2 + 0.00107857*Z2*Z3        E1LG021
     2     - ALOG(Z)                                                    E1LG022
      E1LOG = ALOG(E1)                                                  E1LG023
      RETURN                                                            E1LG024
  100 CONTINUE                                                          E1LG025
C     ARGUMENTS GREATER THAN UNITY                                      E1LG026
      E1LOG = ALOG(Z2*Z2 +  8.5733287401*Z3 + 18.0590169730*Z2          E1LG027
     1       +  8.6347608925*Z  +  0.2677737343 )                       E1LG028
     2       - ALOG(Z2*Z2 +  9.5733223454*Z3 + 25.6329561486*Z2         E1LG029
     3       + 21.0996530827*Z  +  3.9584969228 )                       E1LG030
     4       - ALOG(Z) - Z                                              E1LG031
      RETURN                                                            E1LG032
  200 WRITE(NO,205) Z                                                   E1LG033
  205 FORMAT(6X,'ARGUMENT OF EXPONENTIAL INTEGRAL IS LESS THAN',        E1LG034
     1' OR EQUAL TO ZERO',/,6X,'Z = ',E12.6,3X,'  -- PROGRAM',         E1LG035
     2' TERMINATED')                                                    E1LG036
      STOP                                                              E1LG037
      END                                                               E1LG038
```

```
      FUNCTION FUNCTN(Z)                                                FUNC001
C     INTEGRAND OF HANTUSH WELL FUNCTION                                FUNC002
      REAL*8 DB,Z,ARG,FUNCTN                                            FUNC003
      COMMON BF                                                         FUNC004
      DB=BF                                                             FUNC005
      ARG = DLOG(Z) + Z + DB*DB/(4.0D00*Z)                              FUNC006
      FUNCTN = DEXP(-ARG)                                               FUNC007
      RETURN                                                            FUNC008
      END                                                               FUNC009
```

```
      FUNCTION GAUSS(A,B,FUNCTN)                                        GAUS001
C                                                                       GAUS002
C     NUMERICAL INTEGRATION  BY 24 POINT GAUSS-LEGENDRE QUADRATURE      GAUS003
C     ZEROS AND WEIGHTING FACTORS ARE FROM TABLE 25.4, P916, OF         GAUS004
C     ABRAMOWITZ AND STENGUN(1966)                                      GAUS005
C                                                                       GAUS006
      REAL*8 A,B,C,D,FUNCTN,GAUSS,SUM,W,Z                               GAUS007
      DIMENSION Z(12),W(12)                                             GAUS008
C                                                                       GAUS009
C                                                                       GAUS010
C                                                                       GAUS011
      DATA Z/0.064056892862065,0.191118867473616,0.315042679696163,    GAUS012
     1       0.433793507626045,0.545421471388839,0.648093651936975,    GAUS013
     2       0.740124191578554,0.820001985973902,0.886415527004401,    GAUS014
     3       0.938274552002732,0.974728555971309,0.995187219997021/    GAUS015
C                                                                       GAUS016
      DATA W/0.127938195346752,0.125837456346828,0.121670472927803,    GAUS017
     1       0.115505668053725,0.107444270115965,0.097618652104113,    GAUS018
     2       0.086190161531953,0.073346481411080,0.059298584915436,    GAUS019
     3       0.044277438817419,0.028531388628933,0.012341229799987/    GAUS020
C                                                                       GAUS021
C                                                                       GAUS022
C.....SET UP INITIAL PARAMETERS                                         GAUS023
      C = (B-A)/2.0D00                                                  GAUS024
      D = (B+A)/2.0D00                                                  GAUS025
C                                                                       GAUS026
C.....ACCUMULATE THE SUM IN THE 24-POINT FORMULA                        GAUS027
      SUM = 0.0                                                         GAUS028
      DO 5  J = 1, 12                                                   GAUS029
    IF(Z(J).EQ.0.0) SUM = SUM + W(J)*FUNCTN(D)                          GAUS030
    IF(Z(J).NE.0.0) SUM = SUM + W(J)*(FUNCTN(Z(J)*C + D)                GAUS031
     1              + FUNCTN(-Z(J)*C + D))                             GAUS032
  5  CONTINUE                                                          GAUS033
C                                                                       GAUS034
C.....MAKE INTERVAL CORRECTION AND RETURN                               GAUS035
      GAUSS = C*SUM                                                     GAUS036
      RETURN                                                            GAUS037
      END                                                               GAUS038
```

```
      SUBROUTINE SOL2D(C,PEX,PEY,TSOL,N,NR)                              SOL2001
C     NUMERICAL EVALUATION OF ANALYTICAL SOLUTION                        SOL2002
C                                                                        SOL2003
C     REVISED: 18 APRIL 1984                                             SOL2004
C                                                                        SOL2005
      REAL LAMBDA                                                        SOL2006
      COMMON/RATE/Q(10,12),T(10,12)                                      SOL2007
      COMMON/PHYPRO/ALPHA,BETA,DX,LAMBDA,PE,RD,V                         SOL2008
C                                                                        SOL2009
      PEXY = SQRT(PEX**2 + BETA*(PEY**2))                                SOL2010
      MT = NR + 1                                                        SOL2011
      IF(MT.GT.1) GO TO 10                                               SOL2012
C                                                                        SOL2013
C     STEADY-STATE SOLUTION                                              SOL2014
C                                                                        SOL2015
      S = Q(N,1)                                                         SOL2016
      B = 0.5*PEXY*ALPHA                                                 SOL2017
      C = 2.0*LAMBDA*S*EXP(PEX/2.0 + BKOLOG(B))                          SOL2018
      GO TO 50                                                           SOL2019
C                                                                        SOL2020
C     TRANSIENT SOLUTION                                                 SOL2021
   10 C = 0.0                                                            SOL2022
      IF(T(N,MT).LT.TSOL) MT=MT+1                                        SOL2023
      DO 40  K=2,MT                                                      SOL2024
          IF(T(N,K-1).GT.TSOL) GO TO 50                                  SOL2025
          S = Q(N,K) - Q(N,K-1)                                          SOL2026
          PINJ = V*V*(TSOL-T(N,K-1))/(DX*RD)                             SOL2027
          TAU = PINJ/(PEXY*PEXY)                                         SOL2028
          U = 0.25/TAU                                                   SOL2029
          B = 0.5*PEXY*ALPHA                                             SOL2030
          IF (B.GT.20.0) GO TO 20                                        SOL2031
          WF =   W(U,B)                                                  SOL2032
          SUMLOG = PEX/2.0 + ALOG(WF)                                    SOL2033
          TERM = EXP(SUMLOG)                                             SOL2034
          GO TO 30                                                       SOL2035
C                                                                        SOL2036
C                                                                        SOL2037
C       FOR LARGE VALUES OF B USE THE THIRD ORDER APPROXIMATION          SOL2038
C       OF WILSON AND MILLER (1979) JOUR. HYDRAULICS DIV., ASCE,         SOL2039
C       VOL 105, NO HY12, P 1565.                                        SOL2040
C                                                                        SOL2041
C       NOTE:  THE TERM EXP(PEX/2) IS INCLUDED IN THE NUMERICAL          SOL2042
C       APPROXIMATION FOR W(U,B) TO AVOID COMPUTATIONAL DIFFICULTIES     SOL2043
C       IN TAKING THE PRODUCT  EXP(PEX/2)*W(U,B)                         SOL2044
C                                                                        SOL2045
C                                                                        SOL2046
   20         TERM = WELPRD(U,B,PEX)                                     SOL2047
   30         IF(TERM.LE.1.0E-32) TERM=0.0                               SOL2048
              C = C + LAMBDA*S*TERM                                      SOL2049
   40 CONTINUE                                                           SOL2050
   50 RETURN                                                             SOL2051
      END                                                                SOL2052
```

```
      FUNCTION W(U,B)                                                    WELL001
C                                                                        WELL002
C     JAN WAGNER                                                         WELL003
C     SCHOOL OF CHEMICAL ENGINEERING                                     WELL004
C     OKLAHOMA STATE UNIVERSITY                                          WELL005
C     STILLWATER, OK  74078                                              WELL006
C     TELEPHONE: (405) 624-5280                                          WELL007
C                                                                        WELL008
C     REVISED: 6 JANUARY 1983                                            WELL009
C                                                                        WELL010
C     EVALUATION OF THE WELL FUNCTION FOR LEAKY ARTESIAN AQUIFERS        WELL011
C     THIS VERSION HANDLES ARGUMENTS OVER THE ENTIRE RANGE               WELL012
      REAL*8 A1,A2,FUNCTN,GAUSS,DZ                                       WELL013
      EXTERNAL FUNCTN                                                    WELL014
      COMMON BF                                                          WELL015
      BF=B                                                               WELL016
      IF(B.GT.0.1) GO TO 200                                             WELL017
      IF(U.GT.1.0) GO TO 100                                             WELL018
C                                                                        WELL019
C     FOR B < 0.1  USE APRROXIMATIONS PRESENTED BY                       WELL020
C     HANTUSH, M.S. AND C.E. JACOB (1955)                                WELL021
C     TRANSACTIONS AMERICAN GEOPHYSICAL UNION,                           WELL022
C     VOL 36, NO. 1 PP. 95 - 100.                                        WELL023
C                                                                        WELL024
C                                                                        WELL025
      IF(U.LE.1.0E-10) GO TO 50                                          WELL026
C                                                                        WELL027
C                                                                        WELL028
C     EQUATION 12, FOR U < 1.0                                           WELL029
C                                                                        WELL030
C                                                                        WELL031
      CON = B*B/(4.0*U)                                                  WELL032
      TERM1 = 2.0*BKO(B)                                                 WELL033
      TERM2 = EXP(BIOLOG(B) + E1LOG(CON))                                WELL034
      E1U = EXP(E1LOG(U))                                                WELL035
      SUM = 0.57721566 + ALOG(U) + E1U + (U*B*B/16.0)*(1.0 - U/9.0)      WELL036
      SUMLOG = ALOG(SUM)                                                 WELL037
      TERM3 = EXP(SUMLOG - CON)                                          WELL038
      W = TERM1 - TERM2 + TERM3                                          WELL039
      RETURN                                                             WELL040
   50 W = 2.0*BKO(B)                                                     WELL041
      RETURN                                                             WELL042
C                                                                        WELL043
C     RECIPROCAL RELATION, EQUATION 17, FOR U > 1.0                      WELL044
C                                                                        WELL045
C                                                                        WELL046
  100 TERM1 = EXP(BIOLOG(B) + E1LOG(U))                                  WELL047
      SUM = (B*B/4.0)*(1.0/U - 1.0/(36.0*U*U))                           WELL048
     1       + ((B*B/4.0)**2)*(1.0/(4.0*U) - 1.0/(4.0*U*U))              WELL049
      SUMLOG = ALOG(SUM)                                                 WELL050
      TERM2 = EXP(SUMLOG - U)                                            WELL051
      W = TERM1 - TERM2                                                  WELL052
      RETURN                                                             WELL053
C                                                                        WELL054
C                                                                        WELL055
  200 CONTINUE                                                           WELL056
C     FOR 0.1 < B < 20.0  USE NUMERICAL INTEGRATION                      WELL057
C     FOR U < B/2, W(U,B) = 2KO(B)-INT(O--U) FUNCTION                    WELL058
C     FOR U > B/2  W(U,B) = INT(O--B**2/4U) FUNCTION                     WELL059
C                                                                        WELL060
C                                                                        WELL061
      A1 = 0.0                                                           WELL062
      A2 = U                                                             WELL063
      B2 = B/2.0D00                                                      WELL064
      IF(U.GE.B2) A2=B2*B2/U                                             WELL065
      DZ = GAUSS(A1,A2,FUNCTN)                                           WELL066
      Z = DZ                                                             WELL067
      W = 2.0*BKO(B) - Z                                                 WELL068
      IF(U.GE.B2) W=Z                                                    WELL069
      RETURN                                                             WELL070
```

```
END                                                    WELLO71
```

```
      FUNCTION WELPRD(U,B,PEX)                                          WELP001
C     JAN WAGNER                                                        WELP002
C     SCHOOL OF CHEMICAL ENGINEERING                                    WELP003
C     OKLAHOMA STATE UNIVERSITY                                         WELP004
C     STILLWATER, OK  74078                                             WELP005
C     TELEPHONE: (405) 624-5280                                         WELP006
C                                                                       WELP007
C     REVISED: 6 JANUARY 1983                                           WELP008
C                                                                       WELP009
C     THIS FUNCTION SUBROUTINE EVALUATES EXP(PEX/2)*W(U,B)              WELP010
C     USING THE THIRD-ORDER APPROXIMATION FOR W(U,B) PRESENTED          WELP011
C     BY WILSON AND MILLER (1979) JOUR. HYDRAULICS DIV., ASCE,          WELP012
C     VOL 105,NO HY12, P 1565.                                          WELP013
C                                                                       WELP014
      REAL*8 DI, FZ, TERMI, TERMO, SUM, Z                               WELP015
      COMMON/IO/NI,NO                                                   WELP016
C                                                                       WELP017
      PAR = (B-2.0*U)/((4.0*U)**0.5)                                    WELP018
      IF(ABS(PAR).GT.3.0) GO TO 50                                      WELP019
      TERM1 = (1.0 - 1.0/(8.0*B))*ERFC(-PAR)                            WELP020
      TERM2 = (PAR/(7.0898154*B))/EXP(PAR**2)                           WELP021
C     W = ((1.5707963/B)**0.5)*EXP(-B)*(TERM1+TERM2)                    WELP022
C     WELPRD = EXP(PEX/2)*W(U,B)                                        WELP023
C      = ((1.5707963/B)**0.5)*EXP(PEX/2 - B)*(TERM1+TERM2)              WELP024
      SUMLOG = ALOG(TERM1 + TERM2)                                      WELP025
      WELOG = 0.5*(0.45158271 - ALOG(B)) + (PEX/2.0 - B) + SUMLOG       WELP026
      IF(WELOG.LT.-72.0) GO TO 20                                       WELP027
      WELPRD = EXP(WELOG)                                               WELP028
      RETURN                                                            WELP029
   20 CONTINUE                                                          WELP030
      WELPRD = 1.0E-32                                                  WELP031
      RETURN                                                            WELP032
   50 CONTINUE                                                          WELP033
      IF(PAR.LT.0.0) GO TO 100                                          WELP034
C     FOR B>20 AND PAR>3.0 W(U,B)=2K0(B)                                WELP035
      WELOG = PEX/2.0 + BKOLOG(B) + 0.69314718                          WELP036
      WELPRD = EXP(WELOG)                                               WELP037
      RETURN                                                            WELP038
  100 CONTINUE                                                          WELP039
C     FOR PAR<-3.0 AN ASYMPTOTIC EXPANSION FOR ERFC(-PAR) IS UTILIZED   WELP040
C     SEE SECTION 7.1 OF ABRAMOWITZ AND STEGUN (1966)                   WELP041
      COEFLG = PEX/2.0 - B - PAR*PAR - ALOG(-PAR) - 0.5*ALOG(2.0*B)     WELP042
     1         + ALOG(1.0 - 0.1250/B)                                   WELP043
      IF(COEFLG.LT.-72.0) GO TO 200                                     WELP044
      Z = -PAR                                                          WELP045
      FZ = 2.0D00*Z*Z                                                   WELP046
      SUM = 1.0D00                                                      WELP047
      TERMO = 1.0D00                                                    WELP048
      DO 120  I=2,50                                                    WELP049
          DI = I                                                        WELP050
          TERMI = -TERMO*(2.0D00*DI-3.0D00)/FZ                          WELP051
          IF(DABS(TERMI).GT.DABS(TERMO)) GO TO 150                      WELP052
          SUM = SUM + TERMI                                             WELP053
          TEST = TERMI/SUM                                              WELP054
          IF(ABS(TEST).LT.1.0E-16) GO TO 150                            WELP055
          TERMO = TERMI                                                 WELP056
  120 CONTINUE                                                          WELP057
      WRITE(NO,500)                                                     WELP058
  500 FORMAT(6X,'*** WARNING -- ASYMPTOTIC APPROXIMATION FOR ERFC IN',/, WELP059
     1       6X,'                FUNCTION WELPRD DID NOT CONVERGE WITH ',/ WELP060
     2       6X,'                50 TERMS IN THE SUMMATION')            WELP061
  150 SUMLOG = DLOG(SUM)                                                WELP062
      WELOG = COEFLG + SUMLOG                                           WELP063
      WELPRD = EXP(WELOG)                                               WELP064
      RETURN                                                            WELP065
  200 CONTINUE                                                          WELP066
C     FOR LARGE NEGATIVE VALUES OF PAR, ERFC(-PAR) -> 2                 WELP067
      WELPRD = 0.0                                                      WELP068
      RETURN                                                            WELP069
      END                                                               WELP070
```